Observatory Control and Astronomical Analysis System

for Linux

Reference Manual for OCAAS Version 2.0

Clear Sky Institute

Astronomical Software Specialists

This document created using Applixware 4.4.1 running on Linux kernel 2.0.35.

The most recent version of this document may be found from the CSI home page at www.ClearSkyInstitute.com

Short Contents

Complete Contents

1 Observatory Control and Astronomical Analysis System

1.1 Introduction

OCAAS is a complete UNIX software package for the local and remote operation of an astronomical observatory. It supports both interactive real-time operation and unattended batch-scheduled operation. It controls all aspects of the telescope mount, CCD camera, filter wheel, focuser, weather instrumentation, power supply, GPS receiver, internet or phone line communications, dome and shutter hardware. It automatically performs image corrections, WCS coordinate calibration, compression and data transmission. OCAAS includes programs for off-line photometric and astrometric image analysis of static and time-varying phenomena.

1.2 Main Control Features

1.2.1 Supports equatorial and altitude-azimuth mounts

OCAAS allows the telescope axes to be oriented in any orientation. Thus, equatorial, altitude-azimuth or any mount situation is acceptable. This flexibility accommodates portable operation particularly well. German equatorial mounts are supported. A fast 3-star alignment procedure computes the orientation to good precision, and can be refined to higher accuracy by automatically taking, analyzing and applying a sky mesh. OCAAS accommodates field rotation, which occurs for any orientation other than equatorial, by generating commands to operate a field rotation motor.

1.2.2 Tracks planets, comets, asteroids, and Earth satellites

Ephemerides are included for all planets. Orbital elements for current comets and all numbered asteroids are included and may be easily extended and updated. Earth satellites are fully supported and may be tracked if very current NASA Two-Line-Element sets have been retrieved and the mount is capable of tracking rates on the order of degrees per second.

1.2.3 Unattended scheduled observing

A simple language is used for specifying details of an observation request, such as filter, duration, source name or RA/Dec, offsets, special calibrations, repeat count and delays. Files of requests may be created using any text editor, with a Web page form, or with a GUI application, *mksch*. Collections of individual observing requests are then combined using a scheduling tool, *telsched,* into an efficient interleaved command sequence which will be executed unattended throughout a night by the *telrun* system. Images will automatically be corrected and calibrated in parallel with acquisition. All details of each observation are logged individually, in addition to a continuous log of all engineering data. This automated sequencing may also be interrupted at any time for an unrelated set of requests, then resumed as necessary.

1.2.4 Remote Internet or Phone operation

All control software is fully network-aware using the X Windows protocol. Thus, real-time remote operation and monitoring can be performed using any viable network connection just as easily as operating locally. Remote scheduled operation is a matter of sending a command sequence file before the night begins, then retrieving the images and logs the next morning.

Ethernet connections are preferable, but ISDN and PPP connections to an ISP via modem are also supported. Just be aware of large data sets which can be generated.

1.2.5 Automated powerup and axes calibration sequencing

When power is first applied to the OCAAS computer, it can automatically initiates a basic self-test sequence. Then each axis is sent to find its home position to calibrate the motor step and encoder positions. When these procedures complete, typically in less than two minutes, the system is fully operational and, if batch observations are pending, observing commences (or resumes) automatically.

1.2.6 Simple Basic-Alignment procedure

When a telescope running OCAAS is first installed at a new site, the telescope mount orientation, basic flexure and non-perpendicularity of the principle axes must be determined as a set of descriptor coefficients. This is all accomplished using a simple procedure which requires only that three known stars be located in sequence in an eyepiece or the CCD camera. The operator is prompted during each step of this procedure which takes about 15 minutes to perform. This procedure need only be repeated if the telescope mount is disturbed or modified.

1.2.7 Fine-alignment procedure

Once Basic-Alignment has been completed, the telescope is typically capable of acquiring targets within a few arc minutes of accuracy. If this is acceptable no further alignment is necessary.

However, if better acquisition accuracy is required then a fine-alignment procedure may be performed. This procedure compensates for variety of systematic errors including unusual mount flexure, incorrect location and drive train peculiarities. The first step is to schedule several hundred images to be taken which cover the sky in a fine mesh. This is accomplished using a single menu selection from the *telsched* batch preparation tool and starting the scheduled acquisition system, *telrun*. As with all scheduled acquisitions, these images will be calibrated to sub-arcsecond accuracy. When they are all finished, they are analyzed and combined into a map of pointing errors using the tool *pterrors.* The resulting map is then installed and will be automatically utilized for all subsequent pointing operations.

It usually requires four to six hours to acquire the fine-alignment mesh images, but it does not require operator attention. The procedure need only be repeated if the telescope mount is known to have been disturbed or modified in some way, or if recent images suggest pointing errors have begun to reoccur. Note that uncertainties in atmospheric refraction will generally preclude very high pointing accuracy when working near the horizon.

1.2.8 Automated temperature-compensated focusing

OCAAS can perform an automatic focus procedure at any time under operator control. This procedure acquires a short series of images at several focus positions and analyses each for sharpness. These are then interpolated to compute an optimal focus position. The procedure takes approximately five minutes and requires no operator assistance once it has been initiated from the menu selection. The focus position thus found is logged along with the current ambient air temperature. If this focus procedure is performed at least two different temperatures, OCAAS control software will then use this log to automatically set the focus position based on the air temperature before each image is acquired. The temperatures can also be detected directly on the telescope using small sensors manufactured by Dallas Semiconductor.

1.2.9 All images automatically corrected and WCS calibrated

As OCAAS acquires each raw image from CCD camera, they are automatically corrected with the appropriate bias, thermal and flat frames. This can be true for both real-time operation and scheduled operation. Cataloged reference frames may be used, or new bias and thermal frames may be generated specifically just before each image. Dome flats may also be taken automatically if desired. The details of this processing are always recorded in the FITS header. After corrections have been applied, stars throughout the field are identified and pattern matched to the Hubble Guide Star Catalog to compute a best RA, Dec and field rotation. This position calibration is recorded in the FITS header using the standard World Coordinate System FITS header keywords.

1.2.10 Lossless or Optimized image compression

As images are acquired, they can be automatically compressed to reduce data storage requirements. The compression can be lossless and will achieve approximately 3:1 savings in space. Or a compression algorithm which is optimized for astronomical images can be specified which can achieve file size reductions of 10:1 or more while preserving all quantitative photometric and astrometric characteristics of the images. The algorithm used is known as H-Compression and was built by the Space Telescope Science Institute for managing images from the Hubble Space Telescope. The type and degree of compression can be specified separately for each observation.

1.2.11 Field rotation control (required only for non-equatorial mounts)

Field rotation will occur during any extended exposure for a telescope mount whose polar axis is not aligned very well with the celestial pole. Once at least the basic-alignment procedure has been performed to determine the polar axis orientation, OCAAS control software can compute the field rotation in real-time during each exposure. This can operate a stepper motor attached to the camera to counter rotate and effectively remove the rotation effect from the exposure.

1.2.12 Dome, shutter and roof control

OCAAS can read an incremental encoder attached to a dome, and use the information to control a bi-directional A/C motor to automatically maintain dome slit alignment with the current telescope pointing position. OCAAS can also operate a motor to open and close a shutter curtain on the dome, and pre-rotate the dome to a fixed position each time if necessary to align power take-off wipers for the shutter motor power. Or, the shutter control can be used alone to activate a motor for a roll-off roof.

1.2.13 Continuous Weather monitoring and logging

OCAAS can monitor local meteorological data on a continuous basis and automatically terminate further image acquisition and initiate shutter or roof closing immediately if preconfigured limits for temperature, humidity or wind parameters are reached. When these conditions no longer exist for a configurable period operation will automatically resume. As each image is acquired all weather data are logged in the FITS header. The horizontal and vertical Full-Width-Half-Max statistics for each image are computed and stored in the FITS header to facilitate quantitative investigations of seeing. OCAAS stores all meteorological data to a log file when any parameter changes by a configurable amount. These logs are compact and useful for studying the long-term weather characteristics of a site.

1.2.14 GPS Location and time

OCAAS can continuously monitor a GPS receiver to maintain the system time to much greater than one second accuracy. It can also use the GPS receiver to initialize the geographic location of the telescope. This is particularly handy for mobile applications.

1.3 System Requirements and Assumptions

OCAAS is a software system only. It does not include any hardware, such as a computer, motors, telescope, camera *etc.* The customer must choose and purchase these items separately. The information regarding hardware in this section is intended primarily for planning purposes and for initial engineering designs. Before purchasing decisions are finalized, it is recommended that *Clear Sky Institute* be contacted to insure compatibility and clarify any issues.

Every effort has been made to allow OCAAS to function with many telescopes, domes and cameras. However, it is clearly impossible to guarantee it will work in all cases. CSI works closely with *Torus Precision Optics, Inc.,* and *Apogee Instruments, Inc.,* to insure OCAAS works with their telescopes and CCD cameras, respectively. Choosing these suppliers for these components will eliminate all risks with regards to compatibility with OCAAS software.

1.3.1 Pentium CPU

OCAAS software drivers are currently written for controllers which function in the PC hardware platform with at least two ISA slots. We can report excellent experience with the Dell XPS series systems using a 166 or 200 MHz processor with 32MB of RAM. This system never shows CPU utilization higher than 20% during all phases of image acquisition, correction, calibration and compression.

1.3.2 Linux ELF 2.0.30 or newer

OCAAS software is written in ANSI C. GUI applications are written with the X Windows and Motif API libraries. The software does not use any user process features specific to Linux and is known to function on other UNIX systems, with the exception of the low-level device drivers which have been optimized for Linux. We can report very reliable operation using the Linux 2.0.30 kernel and the Slackware 3.3 distribution. We also know OCAAS works fine under Red Hat release 5.1, which some might prefer for its graphical system administrative tools.

1.3.3 OMS PC-39-E6 Intelligent ISA Motion controller

All motors and incremental encoders interface to the PC hardware using a PC39 intelligent controller built by Oregon Micro Systems. This controller is a full-width card for the ISA bus interface. It has high level commands for features such as cosine profile accelerations, home and limit switch logic, motor pulse generation, encoder pulse quadrature detection and accumulators, and multiple axis control. These facilities of this controller are important for off-loading the time-critical pulse generation from the PC host. This particular model can operate six motors and read two encoders. Other combination are available.

Other controllers, such as those from Galil or Parker, should also be suitable in principle although Linux drivers for them are not yet available.

1.3.4 Stepper (or servo) motors on all axes

OCAAS requires motors to control all axes, including HA (or Az), Dec (or Alt), focus, filter wheel, and field rotator (if necessary). Through suitable amplifiers, the PC39 (mentioned above) can drive either stepper or servo motors with no change in OCAAS software. OCAAS does not perform periodic drive feedback, so friction drives for the principle axes are highly recommended. We have had outstanding performance with servo motors and recommend these in general over stepper motors for the main telescope axles.

1.3.5 Incremental encoders on mount axes

OCAAS requires incremental shaft encoders on the HA (or Az) and Dec (or Alt) axes. This assures accurate slewing acquisition should modest drive slippage occur. We strongly recommend mounting the encoders directly coaxial with each shaft to eliminate the chance of any slippage from intermediate coupling mechanisms. If the encoder resolution is sufficiently high they can also be used as part of the feedback loop during tracking to refine the ideal velocity commands issued to the motors.

1.3.6 Limit switches on principle axes

For safety and peace of mind, limit switches are generally required at each extreme of travel on each axis. The limit switches must operate independently on each axis. The switches should be set to protect the mechanical and safety aspects of the equipment and installation. The limit switches should *not* be used to restrain the observing circumstances for the site, such as minimum altitude, as these are defined and enforced by the software system. When activated, they immediately kill any pulses to the motor on their axis without requiring software cooperation. The limit switches are also used during the initial power up sequence if the telescope was not stowed before loss of power. Limits on the polar axis must be available for each direction of travel. Clever mounting arrangements will be necessary to avoid a wedge-shaped dead zone.

Limit switches are required on the telescope main axes and are recommended on all other motion controls. However, they can be eliminated if absolutely necessary for weight or size reasons if the home switch for an axis is located at one extreme end of travel for that axis and the nominal operating conditions for the axis is never near the other end of the travel. For example, the focus mechanism is often a good candidate for choosing to not include limit switches.

1.3.7 Home switches on all axes

Home switches are required for all axes. These are used to calibrate the stepper motor pulse counts to known physical locations. The location of the home switches is arbitrary, although it can reduce the time to perform the power up sequence if they are positioned to coincide with the desired stow position. In special cases, they can also be located in such a way to eliminate the need for limit switches on a motion control, as mentioned above.

1.3.8 2 GB disk space

OCAAS system software and all databases require approximately 400 MB of disk space. However, even one evening can accumulate on the order of 1 GB of data so a large amount of disk space will be very handy. We have found 2 GB of space to be practical if the data can be reliably off-loaded nightly.

1.3.9 SpectraSource HPC-1 or any Apogee CCD Camera

The current release of OCAAS software includes drivers for the SpectraSource HPC-1 and all CCD cameras produced by Apogee Instruments, Incorporated. These cameras interface to the PC via their own custom ISA card controller cards.

1.3.10 Dome or Roll-Off roof

OCAAS software is capable of controlling either a rotating dome with a motorized shutter, or a roll-off roof observatory configuration. OCAAS software accommodates a rotating dome with a bidirectional AC motor, an incremental azimuth position encoder, a home switch, and limit switches. Since domes rotate rather slowly as a rule, however, it is usually worth the extra cost to use an absolute encoder which eliminates the need to find the home switch on each power up. OCAAS can accommodate either arrangement. The dome shutter requires a bidirectional AC motor. OCAAS is capable of operating a dome shutter which is supplied power via a take-off wiper that requires the dome to first be rotated to a particular azimuth.

A roll-off roof requires three output lines and three input lines. The output lines are asserted to command Open, Close and Stop operations. The input lines listen for Opened, Closed and Error feedback signals. OCAAS software can be configured to remove the Open and Close assertions if the corresponding affirmation does not arrive within a configurable time period. The outputs are always immediately deactivated if the Error input is asserted.

All current OCAAS dome software controls hardware capable of operating only at TTL levels; it is up to the customer to use these to power their motors.

Since domes tend to vary significantly from one installation to another, the OCAAS design allows for a custom dome “plug-in” to be installed on a case-by-case basis.

1.3.11 Peet Bros. Ultimeter 2000 weather station

OCAAS software supports the Ultimeter 2000 weather station from Peet Bros. This system provides data on wind speed and direction, temperature, humidity, air pressure and precipitation via an RS-232 interface. A daemon process runs continuously to update OCAAS on current conditions for its refraction model; saves all statistics to a file for logging purposes; and monitors wind speed, temperature and humidity for alert conditions. If the latter occur, a *Weather Alert* is issued and OCAAS stops observing and closes the dome (or roof). Each FITS image acquired includes fields for all weather statistics.

1.3.12 GPS Receiver

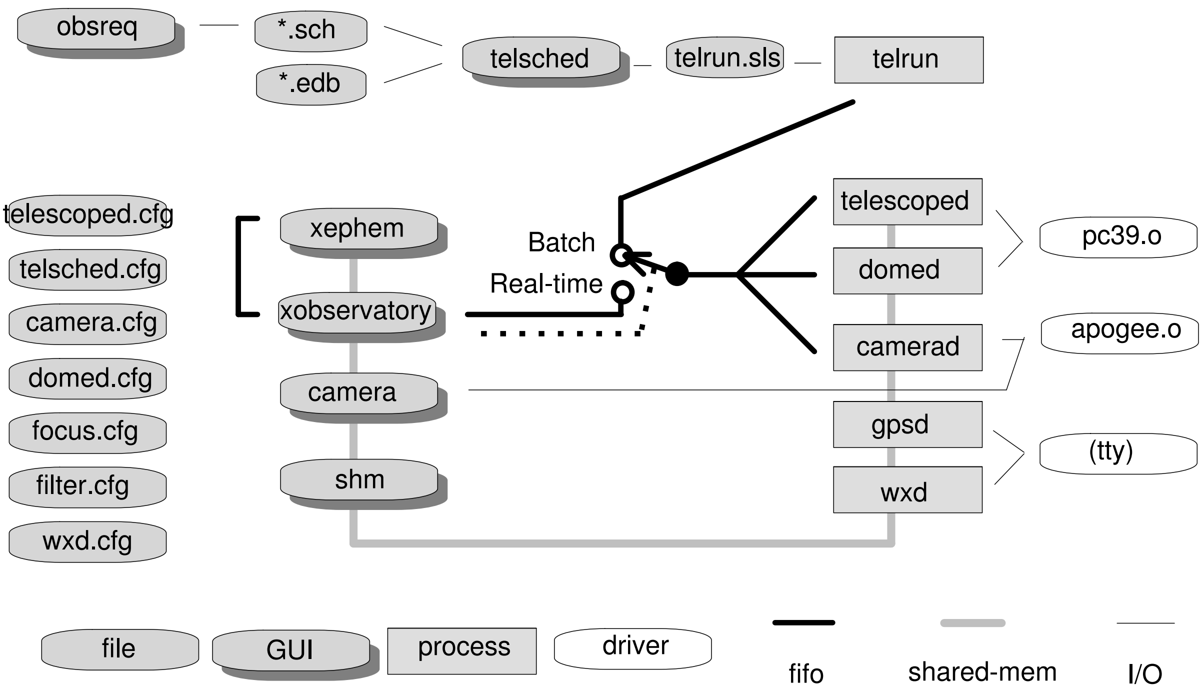
OCAAS can derive time and geographic location from any GPS receiver which can supply the GPRMC NMEA data sentence to a serial port on a timely basis. To date, the Garmin models 45 and 36 have been tested and work well. Other qualifying brands are also very likely to work as well.

1.4 Basic Software Architecture

OCAAS design takes full advantage of the UNIX architecture. It uses long-running *daemon* processes to manage the low-level *drivers* which communicate with the actual hardware. These daemons are accessed using simple text commands via *fifos* (also known as *named pipes*). The commands which flow over these connections are high level and allow the daemons to hide all hardware dependencies. These fifos are used by Graphical User Interface, *GUI*, processes to offer real-time control of the system and by the batch scheduling processes to control the system without direct user intervention. All daemons store their current state in a common *shared memory* segment which processes may use to efficiently learn of current system state and activity.

1.4.1 Data flow diagram

Follows is a diagram which summaries typical data flow through the system. Refer to the diagram during the discussion which follows. This diagram is typical and some systems are built slightly differently. This flexibility is a significant feature of this design.



OCAAS Data Flow

1.4.2 Device Drivers

Starting on the right side of the diagram, the hardware is controlled using three drivers. Two of these drivers, *pc39.o* and *apogee.o* are part of the OCAAS system. These are actually Linux loadable drivers and are installed when the system boots from /etc/rc.d/rc.local.

1.4.2.1 PC39

The stepper motors are controlled via the driver module for the OMS PC39 intelligent motor controller. This driver provides synchronized message-based access to the controller from multiple user processes. The driver supports the *open, write, select, ioctl,* and *close* file operations. The special file to access the driver is $TELHOME/dev/pc39. The module itself is *pc39.o* in the same directory. In the configuration depicted in the diagram, one process, *telescoped,* uses the PC39 to control the motors, home and limit switches on the telescope mount, focus motor and field rotator. Another process, *domed,* uses the PC39 to operate the devices connected to the dome and shutter.

1.4.2.2 Camera

The CCD camera is controlled via a second driver module. This driver provides a uniform interface for all brands of CCD cameras. Functions include:

- setting exposure parameters
- reading the cooler temperature and status
- setting a target cooler temperature
- initiating an exposure
- choice of blocked or polling notification of exposure completion
- reading the pixels from the camera
- aborting an exposure before it completes

Exposure parameters include whether the shutter should be opened, the location of a Subframe, horizontal and vertical pixel binning factors, and exposure duration in seconds. The driver supports the *open, read, select, ioctl,* and *close* file operations. Since this interface is generic across all the cameras supported by OCAAS, application programs use a symbolic link, $TELHOME/dev/ccdcamera, to access the camera. This link points to the special file of the camera driver actually installed. In the configuration depicted in the diagram, the driver is one which operates all models of camera from Apogee Instruments. This driver has a special file named $TELHOME/dev/apogee. The module itself is *apogee.o* in the same directory.

1.4.2.3 Serial ports

The third driver is not really part of OCAAS but is part of the Linux distribution to access RS232 serial ports. These ports are used to connect to various peripherals such a sa weather station, GPS receiver and auxiliary temperature sensors. If the computer does not have sufficient RS232 ports available, a multiport serial card can be used. We can report excellent results using the Stallion EasyIO 4 port multiport board and their driver version 5.3.3.

1.4.3 Daemon processes

UNIX traditionally uses the term *daemon* to refer to processes which are runnable all the while an application is active on the system and which have no direct user interface. Such is the case for all processes described in this section. The OCAAS daemons can be started from the standard Linux system start-up file /etc/rc.d/rc.local which runs the script $TELHOME/archive/config/boot. This script in turn takes instructions from boot.cfg which varies per site depending on the exact configuration and whether the system was configured to be completely autonomous or to have a real-time user interface.

Each daemon listens for relatively high-level commands and issues responses back using several pairs of *fifos* (also known as *named pipes)* for interprocess communication. All fifos reside in $TELHOME/comm. The fifo names end with .in when they are being read by a daemon, and end with .out when being written by a daemon. Also in the comm directory are lock files containing the pid of each daemon and insuring that only one instance of each daemon is running at one time. All fifo traffic is the form of ASCII text. This is very handy for testing because commands can be sent and received using simple UNIX tools such as *echo* and *cat.*

The daemons convert the high-level commands into system-specific commands to suitable device drivers to carry out the command. When the commands are complete, or if errors occur, they issue a response back via these fifos. Thus, fifo messages to the OCAAS daemons are always initiated by a client process which always receives a response. The daemons never initiate communications to other processes via the fifos. In this sense, the daemons may also be referred to as *server* processes.

Note that processes using the fifos are not aware of the architecture of the daemons. There may be one daemon handling each fifo, one handling all or any other combination. This allows the implementation to take advantage of specific operating system and hardware features without effecting application processes. It also allows sites to be uniquely configured by creating daemons specifically matched to, say, an existing dome controller, by just dropping in a new daemon (and possibly new hardware drivers).

All fifo responses are in ASCII and follow the same format: an integer, one space, then a brief English description. The integer 0 always indicates success. Negative values indicate failure. Positive values are used for intermediate progress reports. *These and all subsequent fifo traffic details are subject to change. Contact CSI for the latest detailed list.*

1.4.3.1 telescoped

The *telescoped* daemon is usually responsible for operation of the mount axes, the field rotator, the focuser, and the filter wheel. To perform these duties it uses the *pc39* driver. Basic setup options for telescoped are contained in the configuration file *telescoped.cfg* located in $TELHOME/archive/config. In the same directory are configuration files for setting up the filter, *filter.cfg*, and focus, *focus.cfg*. Only one instance of telescoped may be running at a time, so it leaves a lock file in $TELHOME/comm named *telescoped.pid* which also contains the process id of the daemon.

If the file $TELHOME/archive/config/telescoped.mesh exists, telescoped will use it to refine its pointing accuracy. See the command line program pterrors for more information.

The following table summarizes the syntax of the communications to and from telescoped on each channel.

Channel Name	Input Syntax	Description
Slew	Alt:<rads> Az:<rads E of N>	slew to given Alt/Az and stop
Slew	HA:<rads> Dec:<rads>	slew to given HA/Dec and stop

Channel Name	Input Syntax	Description(cont.)
Track	RA:<rads> Dec:<rads>	slew to given apparent (EOD0 RA/Dec and track
Track	RA:<rads> Dec:<rads> Epoch:<ep>	slew to given astrometric RA/Dec/Epoch and track
Track	<xephem database entry format>	slew to described object and track
Ctrl	Stop	stops all motion immediately
Ctrl	Reset	stops all motion, and rereads all config files
Ctrl	Home	searches for all home switches
Ctrl	Limits	searches for all limit switches and records their encoder values in home.cfg
Focus	<motion, microns>	move focus motor the given signed distance; scale is set in focus.cfg
Filter	<Name>	rotate to center the named filter, as described in filter.cfg
Lights	<n>	intensity, 0 through MAXFLINT. used for dome flats.

1.4.3.2 domed

The *domed* daemon is responsible for operation of the dome. To perform these duties it uses the *pc39* driver. Basic setup options for domed are contained in the configuration file *domed.cfg* located in $TELHOME/archive/config. Only one instance of domed may be running at a time, so it leaves a lock file in $TELHOME/comm named *domed.pid* which also contains the process id of the daemon.

The following table summarizes the syntax of the communications to and from domed on each channel

Channel Name	Input Syntax	Description
Dome	Az:<rads E of N>	rotate to given azimuth and stop
Dome	Auto	maintain slit with telescope without further commands
Dome	Stop	stops all motion immediately
Shutter	Open	open the shutter (set dome az if necessary) or roof
Shutter	Close	close the shutter (set dome az if necessary) or roof
Shutter	Stop	stops all motion immediately

1.4.3.3 camerad

The *camerad* daemon is responsible for batch operation of the camera. (The camera is operated interactively using the Camera program) To communicate with the camera driver it uses the symbolic link $TELHOME/dev/ccdcamera. This is possible because all CCD camera driver modules within OCAAS adhere to the same interface. In the configuration described in the diagram, this is a link to the Apogee driver, *apogee*. Only one instance of camerad may be running at a time, so it leaves a lock file in $TELHOME/comm named *camerad.pid* which also contains the process id of the daemon. Basic setup options for camerad are contained in the configuration file *camera.cfg* (note no d) located in $TELHOME/archive/config.

The following table summarizes the syntax of the communications to and from camerad on each channel. Unlike other channels, there are two response from one Expose command to the Camera channel. The first response indicates that the exposure is complete and the shutter has been closed; the second response indicates the pixels have been read and the camera is free to take another image. This feature allows the telescope to begin slewing to the next target as soon as the camera shutter closes.

Channel Name	**Input Syntax**	**Description**	**Response code**
Camera	Expose ...	Begin an exposure	0 Exposure complete 1 Pixel download complete 3 Camera setup error 4 Readout error
Camera	Stop	aborts the current exposure, if any	2 Aborted (even if none in progress)

1.4.3.4 gpsd

The *gpsd* daemon is responsible for continuous operation of the GPS receiver. To communicate with the receiver it uses the tty specified in *gpsd.cfg*. Only one instance of gpsd may be running at a time, so it leaves a lock file in $TELHOME/comm named *gpsd.pid* which also contains the process id of the daemon.

The gpsd daemon has two roles. On system start-up, it waits to receive a lock from the receiver. Then it reads the geographic location and compares it with that in the configuration file *telsched.cfg.* (This file, like all configuration files, resides in $TELHOME/archive/config.) If the location differs by more than one arc second, it updates the values by appending them to the file. Also on system start-up, it reads the current time from the receiver and sets the Linux clock immediately to this time using the *settimeofday()* system call.

The second role is to maintain system time on an ongoing basis. Gpsd gets the current time from the receiver approximately once per minute. If it differs from the Linux clock, it updates the clock using the *adjtime*() system call. This system call does not abruptly change the time. Rather, it causes a small increase or decrease to the clock rate in a flywheel fashion until the time is correct and then returns to the normal rate. In this way, it adjusts the time and yet maintains the invariant that time always increases monotonically.

Note that neither of these system calls changes the hardware (CMOS) clock. OCAAS does not set the hardware clock. It may be set manually, by root, using the /sbin/hwclock command.

Even though gpsd is not suid-root it has the permission to change the time because it is run from the Linux start-up script /etc/rc.d/rc.local.

1.4.3.5 wxd

The *wxd* daemon is responsible for continual operation of the Peet Bros Ultimeter 2000 weather station. To communicate with the station it uses the tty specified in the *wxd.cfg* configuration file.

Only one instance of wxd may be running at a time, so it leaves a lock file in $TELHOME/comm named *wxd.pid* which also contains the process id of the daemon.

The wxd daemon initializes the serial channel and sets up the Ultimeter 2000 to report all current statistics once per second. It compares each new value with its old value from the previous second. If any value changes by more than a configurable amount it writes a new record of weather data statistics to a file. It also maintains the current weather statistics in a region of shared memory and records a time stamp in shared memory of when the data were last stored. In this way, processes which wish to use the weather data can decide whether the data is stale (should the wxd daemon cease to function for any reason).

The wxd configuration file is *wxd.cfg* located in $TELHOME/archive/config. Each new weather statistics record is appended to the file named *wx.log* located in $TELHOME/archive/logs. This file is reopened, and created if necessary, for append each time a new record is added. Thus, it is fine to move this file out from under wxd for archival purposes. The format of this file is ASCII text, one record per line. Each line is fixed-width with the following format: (the rain total is reset once each 24 hours, although at no particular time of day)

Role	Columns	Units
Julian Date	1-13	Days since Greenwich noon Jan 1, 4713 BC, %13.5f
Wind Speed	15-17	KPH, %3d
Wind Direction	19-21	Degrees E of N, %-3d
Temperature	23-27	degrees C, %5.1f
Humidity	29-31	percent (0..100), %3d
Pressure	33-38	millibars, %6.1f
Rain	40-44	mm since last reset, %5.1f
Weather alert codes	46-50	see below, %5s
Aux temp sensor 1	52-57	degrees C, %6.2f (-99.99 if absent)
Aux temp sensor 2	59-64	degrees C, %6.2f (“)
Aux temp sensor 3	66-71	degrees C, %6.2f (“)

The weather alert column consists of exactly 5 characters. Each character position may be a code if active or a hyphen (-). The possible codes are, in order:

T	Temp is higher than MAXT
C	Temp is lower than MINT
H	Humidity os higher than MAXH
W	Wind speed is higher than MAXWS
R	Rain has increased (any amount)

1.4.4 Directory Structure

All files part of OCAAS are stored beginning at the directory named by the environment variable $TELHOME. This is /usr/local/telescope by default. Follows is a summary of the basic directories under this beginning.

$TELHOME/	**Role**
archive/calib	bias, thermal and flat camera calibration files
archive/catalogs	database files, including *.edb, ppm.xe, gsc
archive/config	all calibration files, such as telsched.cfg, *etc*
archive/images	selected images destined for archival storage
archive/logs	engineering logs kept by all processes
archive/photcal	Landolt photometric standard fields and data
archive/pointmesh	images taken to model telescope pointing errors
archive/telrun	current scheduled command list, in *telrun.sls*
archive/userlogs	staging area for logs while being archived
bin	all executables
comm	communication fifos, lock files
dev	driver modules, their entry points and generic links
user/images	images taken during batch scheduled operation
user/logs	per-schedule engineering log files
user/logs/summary	concise descriptions of each telrun scan list
user/schedin	individual *.sch schedule file submissions
xephem/auxil	support files such as mars and moon images, help
xephem/catalogs	symlink to ../archive/catalogs (for compatibility)
xephem/fifos	symlink to ../comm (for compatibility)

1.4.5 Real-time *vs* Scheduled control

Operation of the observatory facility directly by an operator is accomplished primarily using the *xobs* and *camera* GUI programs. These programs use the various fifos to communicate with the daemon processes to perform each action. They also connect to the shared memory segment to monitor all current state information.

But the fifos can also be connected to *telrun* instead in which case they are handling commands being generated automatically from the scan sequences described in telrun.sls. One basic control available from xobs is to switch Batch operation on and off, which really means to give control of the fifos to telrun (and make sure it is running) or use the fifos directly, respectively.

It is possible to configure an OCAAS system so that when it is booted telrun is started automatically. In this case xobs, and indeed any user interface at all, is optional. If *xobs* is started and it finds *telrun* is already running, it reverts to a passive mode. Or booting can start just *xobs* and be waiting for user input. These configurations are created by suitably modifying the script $TELHOME/archive/config/boot.cfg

2 Installation and Setup

This section discusses planning and installation for a new or updated telescope configuration. For the most part, these topics do not arise during routine operation. The exception is the necessity to find the Home position each time the host computer is powered on, but even this can be configured to be performed automatically.

Follows are the overall steps required to prepare a new installation for OCAAS, in a reasonable order. Some steps may not apply in some situations.

- Prepare observatory building and infrastructure
- Install OCAAS software
- Install and configure computer cards
- Edit configuration files
- Connect GPS and/or set computer time and location
- Align telescope axes using finder or eyepiece
- Focus camera
- Prepare second-order pointing mesh schedule
- Calibrate and confirm dome/roof operation
- Connect weather station

2.1 Preparing the Observatory

A fully operational observatory is much more than OCAAS. OCAAS encompasses only the software aspects of an observatory. Issues such as environmental, electrical, safety, site infrastructure, building, personnel, security, hardware selection and so on are at least equally important but are beyond the scope of this document.

2.1.1 Data Dispensation

One issue which can be discussed is the large amount of data which an automated observatory can generate and the implications this has on storing and retrieving that data.

Consider a 1kx1k CCD camera with 16 bit pixels. Each image will be 2MB. If an average exposure is 60 seconds, one exposure is begun every 2 minutes, and the observing period is 12 hours long, then 360 images will be taken. Assuming the image files are compressed by a lossless method by a factor of two, this will require 360MB of disk storage... each night, night after night. If the observatory is operating at a remote site, this data must be retrieved, either through an electronic transmission or via removable media.

A 56kbit/sec telephone modem operating at 5KB/sec throughput will require 72,000 seconds or 20 hours to transfer this data uncompressed. The slightest glitch and this scheme falls behind. An ISDN connection operating at 100kbit/sec throughput will require 10 hours. At least this will keep up. A 10mbit/sec ethernet operating at a sustained 200KB/sec throughput will require one half hour. This is very reasonable.

An alternative to transmitting the data is to store it on mass media. This will require a small amount of time from a local site operator to attend media changes and arrange for the media to be mailed back to the home facility. This involves operational expenses for several sets of media to be in constant circulation, postage, and operator compensation.

As an example of using current media technology using this scenario, that 360MB of data from one night could be stored on four 100MB Iomega Zip disks at approximately 7MB/$. Or, almost three nights of data could be stored on one 1GB Iomega Jaz disk at about 10MB/$. Or two nights on one CD-RW at about 30MB/$. Or, two nights on one CD-R at about 500MB/$ which is very cost effective indeed even if only a fraction was expected to be archived.

OCAAS installations have been built using all of the above scenarios. Contact CSI for assistance in setting up scripts to manage automated dialing, networking, or media if needed.

2.1.2 Mount Installation

The mechanical issues involved in properly mounting a telescope are many. But once it is firmly mounted, cables are securely positioned (beware of cable wraps at extremes of travel!) and the scope is ready for operation OCAAS includes tools to calibrate the axes and focus the camera. If the telescope is an Alt-Az mount with an image rotator, the reference rotation is also calibrated. There are effectively three methods each of which progressively refine the alignment, but even the initial procedure is often entirely sufficient for nominal tracking accuracy with a good mount.

An exciting implication of fast axis alignment is the possibility of a portable telescope. Mounted on a suitable trailer or other mobile facility, such an instrument would be well suited for rendezvousing with transient phenomena including eclipses, occultations and grazing events.

2.2 Installing OCAAS Software

You must have superuser privilege to perform the installation.

OCAAS for Linux is normally distributed on a CDROM. Begin my mounting it as you would any CDROM on your system. For example, on a Linux system with a drive connected as the master device on the secondary IDE interface, the following command will graft the root of the CDROM file system to /mnt/cdrom:

```
# mount -t iso9660 /dev/hdc /mnt/cdrom
```

Then run the installation script as follows:

```
# /mnt/cdrom/install
```

It will begin by confirming that you wish to install OCAAS. You must then indicate whether you have executed a license agreement to operate OCAAS. If not, the script exits. Next you are asked the directory location for the new software. Installations are encouraged to use the default. If this is an upgrade, your current copy of the GSC catalog will be reused and all images, logs and config files in your current installation may be retained for your reference. Next the script insures there is sufficient disk space to hold the new software. Next the script will check for and create if necessary a new user and group in /etc/passwd and /etc/group, respectively, both named *ocaas.* If this is an upgrade, the existing ocaas login, group and home directory will be preserved.

That ends the questions. The script then copies the software to your hard drive; sets all files to be owned by ocaas user and group; edits /etc/ld.so.conf and executes /sbin/ldconfig to access the OCAAS shared libraries; and adds OCAAS boot capability to /etc/rc.d/rc.local.

2.3 Configuring Controller Cards

The OCAAS PC uses specialized controller cards to communicate with the external hardware. Follows is a discussion of configuring the software to match each of the controller cards supported by OCAAS. Your installation may not use all of these controllers.

2.3.1 Apogee CCD Camera installation

For the Apogee line of CCD cameras, the driver is in $TELHOME/dev/apogee.o. It is actually a Linux module with several parameters which must be specified when installed using the *insmod* utility. The array containing these values is called *apg0.* Most options can be read off directly from corresponding mnemonics in the *.ini* file supplied by Apogee when the camera is purchased. The parameters are as follows:

***apg0* Index**	**Mnemonic**	**Description**
0	base	base address of card, in hex with leading 0x; card occupies 16 bytes
1	rows	total number of actual rows on the CCD chip
2	columns	total number of actual columns on the CCD chip
3	top_border	top rows of the chip to be hidden; must be at least 2
4	bottom_border	bottom rows of the chip to be hidden; must be at least 2
5	left_border	left columns of the chip to be hidden; must be at least 2
6	right_border	right columns of the chip to be hidden; must be at least 2
7	tempcal	temperature calibration, as per .ini file (usually between 100-200)
8	tempscale	temperature scale, as per .ini file but x100 (usually about 210)
9	timescale	time scale, as per .ini file but x100 (usually about 100)
10	caching	1 if the controller supports cached buffer reads, else 0
11	cable	1 if the camera is connected to at least 100 feet of cable, else 0
12	mode	mode value (optional)
13	test	test value (optional)
14	16bit	1 if camera has 16 bit pixels and need to add 32768, 0 if 12 or 14

Another parameter the apogee.o module supports is called *ap_impact.* This value allows you to tune the steadiness with which pixels are read from the camera. Reading pixels at an uneven rate can cause them to change value somewhat and allows later pixels to accumulate more dark current. Tuning is required because the more the CPU is dedicated to reading steadily, the more other unrelated processing is postponed. The higher the value assigned to *ap_impact*, the more steadily the pixels will be read, but the greater "impact" there will be on the rest of the system. A value of 0 has no impact but images may have significant horizontal artifacts if there is any competing CPU load. Increase the value slowly to discover the lowest acceptable values for your system. We find that values of 3 or 4 are usually satisfactory. To read the entire image while completely stopping all other processing, set the value equal to the number of image rows.

An example of a caching AP7 camera on a long cable would be as follows:

```
/sbin/insmod apogee.o apg0=0x290,520,520,4,4,4,4,100,210,100,1,1,4,1,1 ap_impact=3
```

Module loading occurs when the system is booted according to the system configuration script located in $TELHOME/archive/config/boot.cfg. Once installed, the driver is accessed by OCAAS tools such as *camera* via $TELHOME/dev/ccdcamera which in this case should really be a symbolic link to the special file $TELHOME/dev/apogee0.

To test the camera the first time, install the board, run the above command (suitably adjusted for your particular installation), run Camera, bring up the Expose->Setup dialog, and take a test image of 1 second duration with the shutter set to Closed. You should hear the camera shutter open, then close 1 second later. If you do not hear the shutter, the likely cause is no power to the camera. If the camera is connected directly to the computer, all power is drawn from the computer, so shutter and pixels should either both work or neither work. If using the extended cable option, the computer still supplies power for reading the pixels, but the external power supply unit supplies power for the shutter and the on-board thermoelectric cooler.

If the image fails to read and the error *I/O error* is reported by camera, then the likely cause is an intermittent or noisy ground connection. This is especially likely if using the extended cable booster box option. The quick fix is to unplug the 5 pin DIN connector from the external power supply for ten minutes (really) then plug it back in and try again. The correct fix is to use a heavy gauge copper wire and connect the camera connector shield directly to an excellent ground. Also, Apogee now recommends that the computer and the booster box be powered from exactly the same electrical outlet or, even better, the same UPS.

Module loading can occur automatically when the system is booted according to the system configuration script located in $TELHOME/archive/config/boot.cfg.

2.3.1.1 /proc/apogee

Once the apogee driver is installed, current status information is always available from the file /proc/apogee. Use *cat* to read this file and it will produce a list of each apogee control register and a symbolic description of the name of each bit and its current state. This file may be safely read at any time without regards to camera operation.

2.3.1.2 insapogee

By far the easier way to install, or reinstall, the apogee.o camera driver is to use the helper program *insapogee.* This program has three optional arguments and two required arguments. An optional argument is -d *filename* which may be used to specify an alternate driver name; the default is $TELHOME/dev/apogee.o. Optional -v adds additional verbose output. Option -t turns on driver tracing, which causes additional information to be logged to /var/log/messages.

The first required argument is the file name of the original *.ini* file supplied by Apogee Instruments. The second required is the value to be used for the *ap_impact* parameter, as described above. *Insapogee* will also check whether the apogee.o driver is already installed and first uninstall it if found; the driver will not be uninstalled if any program has it in use, such as *camera* or *camerad.*

For example, to install the default driver using the file ap7.ini in $TELHOME/archive/config with an impact value of 3:

```
insapogee -v $TELHOME/archive/config/ap7.ini 3
```

You must be effectively root to run *insapogee*.

2.3.2 PC39 Installation

The OMS PC39 stepper motor driver module is in $TELHOME/dev/pc39.o. It has only one parameter, *pc39_io_base.* This must be set to the base address configured on the controller card. For a card jumpered at address 360 hex for example, the module would be installed with the following command:

```
insmod pc39.o pc39_io_base=0x360
```

The PC39 occupies 4 bytes of I/O address space and does not use interrupts.

Module loading can occur automatically when the system is booted according to the system configuration script located in $TELHOME/archive/config/boot.cfg.

2.3.2.1 Axis assignments

Assignment of PC39 axes to telescope motors is not fixed, but is defined in the appropriate configuration files: telescoped.cfg, focus.cfg and filter.cfg.

2.3.2.2 I/O Port assignments

14 user-definable I/O ports are available on the PC39 in addition to its motor and encoder features. These are assigned as follows. All inputs are low=ground=true; all outputs are low=ground=active. Refer to the OMS documentation for hardware jumpers, pinouts and operating conditions.

It is the customer's responsibility to operate the PC39 within its stated limits.

Pin	Usage
Input 0	Dome shutter (or roof) motor is active
Input 1	Dome home switch
Input 2	Emergency dome/shutter stop
Input 3	Reserved
Input 4	Reserved
Input 5	Reserved
Input 6	Reserved

Pin	Usage(cont.)
Input 7	Reserved
Output 8	Turn on low-intensity dome-flat light
Output 9	Turn on high-intensity dome-flat light
Output 10	Rotate dome CW (as seen from above)
Output 11	Rotate dome CCW (as seen from above)
Output 12	Command dome shutter (or roof) to open
Output 13	Command dome shutter (or roof) to close

These assignments are subject to change and may be effected by settings in the dome.cfg configuration file. Contact CSI before committing to a particular electrical design.

2.3.2.3 /proc/pc39

Once the pc39 driver is installed, the file /proc/pc39 allows reading the last several commands which were sent to the controller and all of its responses. Each *cat* of this file drains and resets the history queue. The queue is only so big (512 bytes as of driver version 2.1) so if the pc39 is very active many commands can be missed since only the most recent are maintained in the history queue.

2.3.3 SpectraSource HPC-1 CCD Camera Installation

The HPC driver module is in $TELHOME/dev/hpc.o. It has two parameters, *hpc0* and *impact.* Hpc0 is the base io address of the card. Impact is the degree to which pixels will be read steadily but at the expense of other system activity. See the description of the Apogee camera driver for more discussion regarding impact because it is the same idea here. For example, a card jumpered at address 120 hex and impact, set to 3 the module could be installed with the following command:

```
insmod hpc.o hpc0=0x120 hpc_impact=3
```

Module loading can occur automatically when the system is booted according to the system configuration script located in $TELHOME/archive/config/boot.cfg. Once installed, the driver is accessed by OCAAS tools such as *camera* via $TELHOME/dev/ccdcamera which in this case should really be a symbolic link to the special file $TELHOME/dev/hpc.

2.3.4 Stallion EasyIO 4-port serial mux

If additional serial ports are needed for OCAAS devices, such as a weather station or GPS receiver, CSI can assist with software and installation for the EasyIO serial port expansion card from Stallion. Other similar cards are supported by most Linux systems and probably work as well but CSI has had good experience and support with the Stallion products. Do *not* use the Stallion driver which is included with most Linux kernel distributions.

2.4 Editing Configuration Files

Several text files are used to define local observing circumstances and specifications of the observatory equipment. These files may be edited using any text editor, such as *vi*. When OCAAS is delivered as part of a total system, these files will already be set and little or no editing will be required. The following sections describe the fields in each file and hints to help determine their correct values if necessary. All config files reside in the directory $TELHOME/archive/config.

The overall structure of each configuration file is the same. There is one parameter per line. The name of the parameter is followed by an optional equals sign (=); then the value; then an optional comment which begins at the first exclamation point (!) and continues to the end of the line. For example:

```
HSTEP        31176565       ! number of motor steps per full revolution
```

A parameter may appear any number of times in a config file, but *only the last will be used*. This characteristic is used intentionally by the software to append an updated value yet leave a trail of previous values.

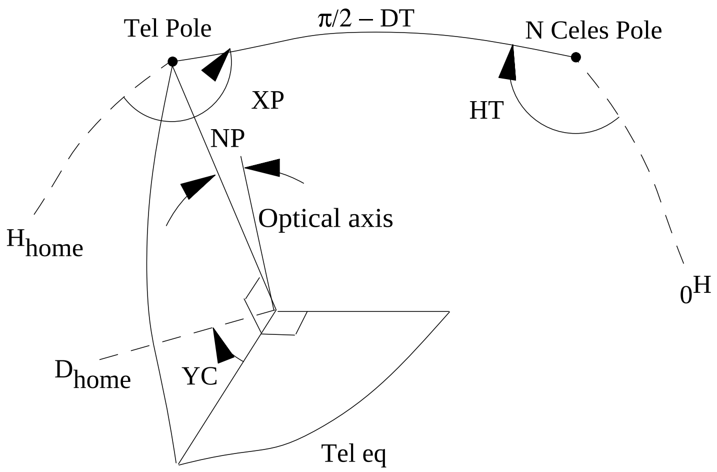
☠ ***It must be strongly emphasized that these files are critical to proper operation. Errors in the configuration files can render the entire system inoperable or even dangerous to equipment and personnel. Again, once properly set, it is seldom if ever necessary to edit these files.***

2.4.1 Canonical Coordinate System

All configuration files refer to the various motions of the telescope using a *canonical coordinate system.* This is to allow consistent thinking regardless of whether the motors on any one axis happen to move in a certain direction when driven with a positive or negative motion command from the hardware controller perspective. Follows is a table which summaries the canonical sign conventions:

Axis or Motor	**Canonical sign convention for *Positive* motion**
HA(Az)	ccw looking down at scope from North (above)
Dec(Alt)	rotating towards scope's North (vertical) pole
Field rotator	cw rotation, as seen looking "through" the camera
Focus	moving to shorten the optical path from primary to camera
Filter wheel	cw rotation, as seen looking "through" the camera
Dome	cw rotation, looking down from above

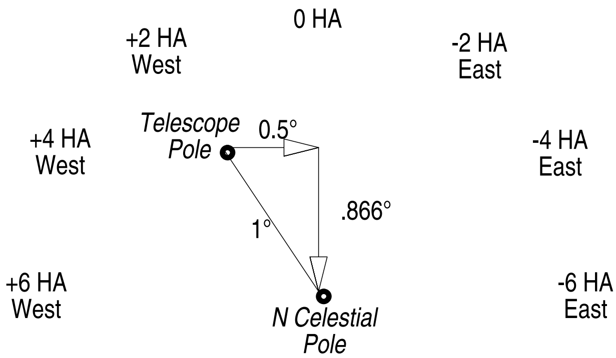
Also, a canonical pointing model is computed which defines the orientation of the telescope in space. The symbols are defined and sketched below:



HT	Hour angle of the North pole of the telescope, in H:M:S
DT	Declination of the North pole of the telescope, in D:M:S
XP	Angle swept from the home switch on the longitudinal axis (HA or Az) to the North celestial pole, moving ccw as looking down from the North (top) end of the telescope, in D:M:S
YC	Angle from the equatorial plane of the telescope to the home switch on the latitudinal axis (Dec or Alt), moving from the plane Northwards (up), in D:M:S
NP	Non-perpendicularity of axes, that is, the deviation from a perfect 90° angle between the latitudinal and polar axes, in D:M:S
R0	The offset for the image rotator, in D:M:S. This value is zero if no rotator is installed.

Note that the values for HT and DT indicate the direction of the telescope's polar axis. For an equatorial telescope, this indicates directly the deviation from perfect polar alignment. Looking at the North celestial pole with the unaided eye, 0 hours Hour Angle is upward from the pole, -6 hours is to the right (east), +6 is to the left (west) and +/- 12 hours is straight down. Thus, the HT value indicates the direction of the telescope pole from the celestial pole. DT is the declination of the telescope's pole and so this value subtracted from 90 degrees is the angular distance of the telescope pole from the celestial pole. A perfect alignment is indicated with a value for DT of exactly 90:0:0. If this is achieved, the value for HT is meaningless.

> For example, suppose HA is reported as 2:0:0 and DT as 89:0:0. This means the telescope pole is pointing 1 degree away from the North celestial pole towards the northwest (when facing north). Since 2 hours is 30 degrees of angle, the 1 degree offset can be decomposed into 0.5 degrees too far west and 0.866 degrees too far up. The mount wedge should be adjusted to correct these errors and the basic alignment procedure repeated.



2.4.2 Using *xedit*

For those with no experience with any UNIX editor, a simplistic but easy-to-use text editor for editing configuration files called *xedit* is available on most Linux systems and will be introduced here. To use this editor, use and xterm window and type *xedit.* This will start a new instance of the xedit application. Click in the space to the right of Load and type the name of the file to be edited. Then click on Load. The file to be edited may also be specified as an argument when xedit is started.

Xedit divides itself into four work areas, or *panes*. Each is separated by *a sash* which can be adjusted by dragging the black square towards the right end of each sash. The top three panes are for various messages so look here for additional information if things do not seem to be working as expected.

The file to be edited is displayed in the lower pane. The current text insertion point is always marked with a small subscript circumflex (∧). Text can be entered at this location by just typing. The insertion point may be moved by using the arrow keys on the keyboard, or placed directly by clicking at the desired location with the mouse. The backspace key deletes one character to the right. To joint two lines, place the insertion point at the beginning of a line and press Backspace.

A scrollbar will appear along the left edge if the file is too long to be shown completely. Clicking the left mouse button while the cursor is in this scrollbar region will scroll down by one page; the right button will scroll up one page. The middle button will allow dragging the file to any desired position.

Edit the file as desired, using the mouse or arrow keys to move around and the Backspace key to delete the character to the left of the insertion cursor. To quit without saving any changes, click on Quit, twice. To save the file with changes, click on Save, then Quit to quit or type a new file and click on Load as before.

For more information on xedit, use an xterm window and type ‘man xedit’.

2.4.3 telescoped.cfg

The file *telescoped.cfg* contains information about the configuration of the telescope mount. The following is a typical example, followed by extra discussion and tips. Throughout the list, a leading “H” in names refers to the longitudinal axis; on an equatorial scope this is the Hour Angle or Polar axis and on an Alt-Az scope this is the Azimuth axis. Similarly, a “D” refers to the latitudinal axis; or Declination or Altitude axes respectively.

```
! CSI development platform

! Main axes calibration constants which are just filled in once forever.
! N.B. home.cfg has more which must be set once but are updated dynamical
! N.B. use ‘dynamics’ tool to find best settings of MAXVEL/MAXACC/POLL_PE
! N.B. all measures and directions are canonical unless stated as raw.
!   HA(Az):     +ccw looking down at scope from North (above) (like RA, r
!   Dec(Alt):   +moving towards scope’s pole (like Dec)
!   Rotator:    +cw rotation, looking "through" the camera


! "H" refers to the longitudinal axis, ie, ha or az.
! "D" refers to the latitudinal axis, ie, dec or alt.
! "R" refers to the field rotator -- must be present but ignored if don’t

HAXIS       x           ! PC39 motor axis
HHAVE       1           ! 1 if H axis is to be active, 0 if not
HPOSSIDE    1           ! 1 if home is side hit first when going pos
HHOMELOW    1           ! 1 if home switch is active low, else 0
HENCHOME    0           ! 1 if use encoder home, 0 for separate switch
HESTEP      8000        ! raw encoder counts/rev
HESIGN      1           ! cnts [1=incr -1=decr] ccw looking down from N
HLIMMARG    .04         ! limit backoff safety margin, rads
HMAXVEL     0.4         ! max velocity, rads/sec
HMAXACC     0.6         ! max acceleration, rads/sec/sec
HSLIMACC    10          ! soft limit and urgent acc, rads/sec/sec
HTRENCWT    0           ! tracking encoder weight: 0(ideal)..1(rd enc)
HDAMP       .75         ! closed-loop damping factor, 0..1

DAXIS       y           ! PC39 motor axis
DHAVE       1           ! 1 if D axis is to be active, 0 if not
DPOSSIDE    1           ! 1 if home is side hit first when going pos
DHOMELOW    0           ! 1 if home switch is active low, else 0
DENCHOME    0           ! 1 if use encoder home, 0 for separate switch
DESTEP      8000        ! raw encoder counts/rev
DESIGN      1           ! steps [1=inc -1=dec] with lat
DLIMMARG    .04         ! limit backoff safety margin, rads
DMAXVEL     0.4         ! max velocity, rads/sec
DMAXACC     0.6         ! max cceleration, rads/sec/sec
DSLIMACC    10          ! soft limit and urgent acc, rads/sec/sec
DTRENCWT    0           ! tracking encoder weight: 0(ideal)..1(rd enc)
DDAMP       .75         ! closed-loop damping factor, 0..1

! field rotator axis calibration constants
RAXIS       z           ! PC39 motor axis
RHAVE       0           ! 1 if even have an image rotator, 0 if not
RHASLIM     0           ! 1 if this axis uses limit switches, else 0.
RPOSSIDE    1           ! 1 if home is side hit first when going pos
RHOMELOW    1           ! 1 if home switch is active low, else 0
RSTEP 9216000           ! raw motor usteps/rev
RSIGN       -1          ! steps [-1=inc 1=dec] to rotate star image cw
RLIMMARG    .01         ! limit backoff safety margin, rads
RMAXVEL     0.349       ! max velocity, rads/sec
RMAXACC     0.3         ! max acceleration, rads/sec/sec
RSLIMACC    2           ! soft limit and urgent acc, rads/sec/sec
RDAMP       .9          ! closed-loop damping factor, 0..1

! misc
POLL_PERIOD 100         ! nominal period between updates, ms
TRACKACC    .01         ! max tracking error, rads, or 0 for 1 enc step
GERMEQ      0           ! 1 if mount is German Equatroial, else 0.
ZENFLIP     0           ! 1 to change alt/az reference side, else 0.
FGUIDEVEL   .001454     ! fine guiding velocity, rads/sec
CGUIDEVEL   .01454      ! coarse jogging velocity, rads/sec
MAXFLINT    3           ! max flat light intensity. 0 for none.

! For RCS Only -- Do Not Edit
```

The first group of parameters refers to the longitudinal axis. **HAXIS** assigns the PC39 axis connected to the motor on this axis. **HHAVE** is set to 1 if the axis is to be controlled by the system or ignored; this is only useful during debugging. **HPOSSIDE** determines which side of the home switch will be used. It is important to always approach a home switch from the same direction and use the same side to achieve maximum repeatability. If this parameter is set to 1the side which is encountered first when the mount axis is moving in the canonical positive direction will be used; setting to 0 will use the other side. **HHOMELOW** allows selecting the polarity of the home switch. A value of 1 means the home voltage goes low when active, 0 means the voltage goes high when active. This also allows using either a normally-open or normally-closed home switch. **HENCHOME** is set to 1 if the encoder has a built-in ability to generate a home pulse, or set to 0 when using an external home switch. Note that when using the internal encoder home, the pc39 home input must be strapped to ground and HHOMELOW must be set to 0. **HESTEP** is the number of encoder pulses generated by one full revolution of this axis. **HESIGN** is set to 1 if the actual hardware sign of the encoder is the same as the canonical direction for this axis, or set to 0 if it is the opposite. **HLIMMARG** is the angular amount by which the software will avoid the hard limits. This is to allow for limit switches which do not repeat precisely. **HMAXVEL** is the maximum velocity to use while slewing to a new object. **HMAXACC** is the maximum acceleration to use during normal operation, such as slewing to a new object. **HSLIMACC** is the acceleration to use when a limit switch is encountered or other severe condition is detected. It should be as high as the hardware can tolerate on an occasional basis. Setting this value too large can result in drive train damage due to the substantial forces which may be applied to achieve the acceleration. Values too small may result in too much travel beyond the limit switch and do damage to other components, or even travel past the limit intended for the other direction and cause great confusion. **HTRENCWT** allows controlling whether or not tracking uses encoder feedback. If the value is 0, then no feedback is used and the tracking velocities are computed each update period assuming no slippage from the motors. If the value is 1, the motor velocities are computed each update period based entirely on the current position as read from the encoder. Values in between will compute a weighted sum of the two values. As a general rule, this value should be set to 0 unless the encoder resolution is at least as high as the motors. **HDAMP** is a feedback loop which effects the dynamical behavior of the software feedback loop during slewing. The value must be between 0 and 1. Values too small will result in underdamping, and the scope will tend to go beyond the target and have to hunt back. Values too large will result in overdamping, and the scope will approach a target very cautiously and tend to take longer than necessary to achieve lock. Beware that any motor slippage can also give the appearance of underdamping and should be reduced or eliminated as much as possible mechanically before trying to tune this value. Also, see the description of the *dynamics* tool which helps explore the effects of this value and assist to set an optimum value.

The next group of parameters is essentially identical but applies to the latitudinal axis.

The next group of parameter applies to the field rotator, if present. The only difference to note is that **RSTEP** and **RSIGN** refer to the motor itself; it is assumed that the field rotator does not utilize an encoder.

In the final section are some miscellaneous parameters, most of which effect the telescope pointing control in general. **POLL_PERIOD** sets the interval between which new drive rates are computed and issued during tracking. For an equatorial mount, which only moves one axis and that very steadily, about 500 ms is adequate. For an Alt-Az mount where three axes are constantly changing, values on the order of 60-80ms are suggested. **TRACKACC** sets the permissible difference between the motor and encoder values while tracking an object (but see **HTRENCWT)** . This should never be set smaller than the worst encoder precision of either axis; as a convenience, setting this to 0 will set it exactly to the precision of the worst encoder. **GERMEQ** is set to 1 for a German Equatorial mount, else set to 0. Several issues related to operating a GE mount with OCAAS are discussed in a separate section. **ZENFLIP** is intended for use with Alt-Az mounts. Setting it to 1 will cause the altitude axis to flip and the scope will operate on the other side of vertical; setting to 0 will use the standard side. **FGUIDEVEL** sets the velocity which the scope will move while tracking and using the Fine controls of the Paddle. **CGUIDEVEL** sets the velocity the scope will move while tracking and using the Coarse controls of the Paddle; this is also the velocity the scope will move to slew (not tracking) and using the Fine controls of the Paddle (when not tracking, the Coarse controls of the Paddle move at the MAXVEL velocity for the given axis).

MAXFLINT sets the maximum brightness level for the dome flat lighting system. This will allow brightness levels in the filter.cfg entries for flat fields to range from 1 up to and including this number. In the standard OCAAS configuration, there are two lights, the first of which is assumed to be dimmer than the second. See the PC39 pinouts for the exact control connections. In this case, MAXFLINT should be 3 to allow brightness levels to be 1, 2 or 3 for dim, bright and both lights on. Set this to 0 if there is no flat light mechanism present.

2.4.4 focus.cfg

This file contains information about the focus hardware. It also contains basic parameters used by *xobs* when performing the automatic focus procedure. Follows is a typical example of this file.

```
! details of the focus motor -- CSI development platform

! basic description
! N.B. see filter.cfg for focus settings per filter.
! canonical positive it towards the main mirror.

OAXIS       z           ! pc39 focus axis
OHAVE       1           ! 1 if even have a focus motor, 0 if not.
OHASLIM     0           ! 1 if this axis uses limit switches, else 0.
OPOSSIDE    1           ! 1 to use the far side of home sw, else 0
OHOMELOW    1           ! 1 if home switch is active low, else 0
OSTEP       3300        ! steps/rev
OSIGN       1           ! 1 if raw pos moves towards camera, -1 if away
OLIMMARG    0           ! limit backoff safety margin, rads
OMAXVEL     0.7         ! max focus motor velocity, rads/sec
OMAXACC     20          ! max focus motor acceleration, rads/sec/sec
OSLIMACC    100         ! soft limit and urgent acc, rads/sec/sec
OSCALE      8           ! steps per micron
OJOGF       1           ! fraction of OMAXVEL when using paddle

! how to perform autofocus
OFIRSTSTEP  250         ! initial movement to star search, microns
OSTOPSTEP   25          ! target focus depth, microns
OEXPTIM     5           ! exposure time for focus cal images, secs
OTRACK      1           ! 1 to insist on scope actively tracking, else 0
OMINSTD           10          ! minimum image std dev we will allow

! For RCS Only -- Do Not Edit
! @(#) $RCSfile: focus.cfg,v $ $Date: 1998/09/28 23:00:49 $ $Revision: 1.
$
```

The first group of parameters defines the basic focuser mechanism. **OAXIS** defines which PC39 axis controls the focus motor. **OHAVE** is set to 1 unless there is no focus motor on the telescope at all. **OHASLIM** is set to 1 if the hardware includes limit switches, else 0 if travel limits are specified entirely in software via the *home.cfg* config file. **OPOSSIDE** is set to 1 if the home switch is to be approached while moving the focuser in the canonical positive direction (that is, towards the camera), 0 if to be approached while moving in the negative direction. It is important to choose the side of the home switch that is most accurate, and to allow for the expected side on which the focuser will normally operate. **OHOMELOW** allows selecting the polarity of the home switch. A value of 1 means the home voltage goes low when active, 0 means the voltage goes high when active. This also allows using either a normally-open or normally-closed home switch. **OSTEP** is the number of motor steps per revolution of the focus screw. **OSIGN** is set to 1 if a raw positive motor step command moves the focuser closer to the camera, or set to 0 if positive moves away. **OLIMMARG** is the angular amount by which the software will avoid the hard limits, if any. This is to allow for limit switches which do not repeat precisely. **OMAXVEL** is the maximum velocity to use while moving to a new focus position. **OMAXACC** is the maximum acceleration to use while moving the focuser in normal usage. **OSLIMACC** is the acceleration to use when a limit switch is encountered or other severe condition is detected. See also the discussion for HSLIMACC. **OSCALE** is the basic scale of the focus mechanism, in units of steps per micron (micrometer) of motion. (It might be seen as a little awkward, but even linear focus mechanisms can be described using this and OSTEP.) **OJOGF** specifies the speed which the focuser moves when being controlled from the Paddle, as a fraction of OMAXVEL. It allows for rapid changes of focus during normal operation, but fine control when experimenting with the Paddle.

The second section defines the basic constraints used during the automatic focus procedure provided by *xobs.* **OFIRSTSTEP** is the number of microns for the first move of the procedure. This should be large enough to definitely effect the focus, but not so large as to hopelessly defocus everything. Suggested values are 5-10 times the depth-of-focus of the optical system. **OSTOPSTEP** sets the amount of motion smaller than which there is no appreciable change in focus quality; this is basically the depth-of-field of the optical system. The procedure stops when it predicts a move less than this is required to further refine the focus. **OEXPTIM** is the default exposure time for each focus test image. This need only be long enough to reliably capture several bright stars, and short exposures will expedite the procedure. **OTRACK** sets whether it will be enforced that the telescope be tracking during the focus procedure. This can generally be set to 0, although it is theoretically optimum to perform the focus procedure on a static scene. **OMINSTD** is the minimum required standard deviation of the pixels in each image to be used for the procedure. This is a sanity check to assure there really are stars in the image, is not just clouds or the dome wall, or grossly over or under exposed.

2.4.5 dome.cfg

This file contains information about the dome or roof configuration.

```
! CSI

! dome constants
DOMEHAVE = 1            ! 1 to enable rotating dome control, 0 to disable
DOMEAXIS = u            ! pc39 encoder axis to use, 0 for dio
DOMETOL = 0.04          ! max dome position tolerance, rads
DOMETO = 90             ! max time to wait for dome to get into position,
DOMEZERO = -0.8726      ! az when home, rads +E of N
DOMESTEP = 1024         ! encoder steps/rev
DOMESIGN = 1            ! 1 if raw encoder increases +E of N, else -1
DOMEHOMELOW = 1         ! 1 if home switch is active low, else 0
DOMEPOSSIDE = 1         ! 1 if home is side hit first when going pos
DOMEMOFFSET = .2        ! (mount offset)/(dome radius), + west or - east

! shutter constants
SHUTTERHAVE = 1         ! 1 to control shutter/roof, 0 if none.
SHUTTERTO = 90          ! max time to wait for shutter to open or close,
SHUTTERFB = 1           ! 1 if shutter has active feedback, 0 to just tim
SHUTTERAZ = 1.39917     ! dome az to contact shutter power rings, rads +E
SHUTTERAZTOL = .04! max dome az power rings tolerance, rads

! setup for dome flats -- FLATDAZ is ignored if no dome.
FLATTAZ = 3.14          ! telescope azimuth for dome flat
FLATTALT = .1           ! telescope altitude for dome flat
FLATDAZ = .3            ! dome azimuth for dome flat

! For RCS Only -- Do Not Edit
! @(#) $RCSfile: dome.cfg,v $ $Date: 1998/06/23 22:54:03 $ $Revision: 1.1
$
```

The first group of parameters describes the rotating dome. **DOMEHAVE** is set to 1unless there is no rotating dome. **DOMEAXIS** is the PC39 channel connected to the dome encoder. **DOMETOL** is the required accuracy of the dome azimuth. Too small and the dome will likely hunt back and forth due to its large mass and tendency to coast; too large and the telescope may not always peer out through the shutter slit. **DOMETO** is the maximum number of seconds any dome rotation command should require. **DOMEZERO** is the azimuth of the dome slit when the home switch is triggered. **DOMESTEP** is the total number of encoder steps for one full revolution. **DOMESIGN** is 1 if the encoder values increase as the dome rotates clockwise (as soon from above looking down), or 0 if the values decrease. **DOMEHOMELOW** is set to 1 if the electronics used to implement the home switch cause it to be grounded when the switch is activated, else 0 if it goes to +5 when activated. **DOMEPOSSIDE** is set to 1 if homing should approach the home switch while rotating clockwise, else 0 to approach rotating counterclockwise.

See the section on roof installation for a discussion of **DOMEMOFFSET.**

The second group of parameters describes the shutter or, if no dome, the roll-off roof. **SHUTTERHAVE** is set to 1 if there is indeed a shutter or roll-off roof connected, else 0. **SHUTTERTO** is the maximum number of seconds a shutter close or open operation should require. **SHUTTERFB** is set to 1 if the electronics can get actual feedback that the shutter is moving. If this is set to 0, then shutter operations are just initiated and then assumed to complete correctly after always waiting for SHUTTERTO has elapsed**. SHUTTERAZ** is the azimuth at which the dome must be positioned to get power to the shutter. **SHUTTERAZTOL** is the accuracy to which the dome must be positioned at SHUTTERAZ to work.

The last group of parameters defines information for setting up the telescope and dome for taking image calibration flats. **FLATTAZ** and **FLATTALT** are the telescope azimuth and altitude to point to the flat. **FLATDAZ** is the dome azimuth setting to properly position the flat. In all cases, the shutter or roof is closed while taking flats.

2.4.6 filter.cfg

This file contains information about the filter hardware, including the number and names of each filter wheel position. It can also capture up to two focus/temperature pairs for each filter, used for temperature compensated focusing, and the desired light setting when taking a dome flat for each filter. Follows is a typical example and further explanation.

```
! Details about the filter wheel motor
IAXIS       v           ! pc39 motor axis

! If using the Torus filter, set following to 1 and ignore the remaining
IHAVETORUS  0           !

IHAVE       0           ! 1 for filter wheel control, 0 to disable.
IHASLIM     0           ! 1 if this axis uses limit switches, else 0.
IPOSSIDE    1           ! 1 to use the cw side of home sw, else 0
IHOMELOW    1           ! 1 if home switch is active low, else 0
ISTEP       9216000     ! usteps per full revolution (or tray travel)
ISIGN       1           ! 1 if raw pos moves cw, -1 if ccw
I1STEP      768000      ! usteps between each filter position
IOFFSET     0           ! usteps from home to first filter center
ILIMMARG    .1          ! limit backoff safety margin (if limits), rads
IMAXVEL     1           ! max focus motor velocity, rads/sec
IMAXACC     6           ! max focus motor acceleration, rads/sec/sec
ISLIMACC    1           ! soft limit and urgent acc, rads/sec/sec

! Details about each filter. Each entry consists of up to 5 fields, separ
! by commas (,) as follows, (each also lists a default if missing):
!
!   Name, must begin with a unique letter, case is ignored (required)
!   flat calibration time, seconds (FLATDURDEF)
!   flat light source intensity, 1..3     (FLATLTEDEF)
!   focus setting 1 as "microns_from_home/temperature_C" (NOMPOSDEF/FILTT
!   focus setting 2 as "microns_from_home/temperature_C" (setting 1)
!

FLATDURDEF  5           ! default flat duration, secs
FLATLTEDEF  1           ! default flat illumination level, 1..3
NOMPOSDEF   0           ! default focus position, microns from home
FILTTDEF    0           ! default temperature, C

FILT0       'Mirror'
FILT1       'Clear,10,1,4543/10,4555/-10'
FILT2       'Blue'
FILT3       'Visual'
FILT4       'Red'
FILT5       'Infrared'
FILT6       'W H Alpha'
FILT7       'X H Off'
FILT8       'Y H Beta'
FILT9       'Z Oxygen III'
FILT10      'No Filter'
FILT11      'P'

NFILT       12          ! total number of filter positions

FDEFLT      'Clear'     ! default filter after homing

! N.B. Even if IHAVE is set to 0, FDEFLT must still be defined and will b
! used to store flat and focus info. FILT* and NFILT must also be sensibl

! For RCS Only -- Do Not Edit
! @(#) $RCSfile: filter.cfg,v $ $Date: 1998/06/23 22:54:03 $ $Revision: 1
$

! Entries beyond this point were filled in by the software
```

The first group of parameters defines the basic filter mechanism. **IAXIS** defines which PC39 axis controls the filter motor. If you are using the Torus filter wheel, set **IHAVETORUS** and ignore the remaining values in this section. **IHAVE** is set to 1 unless there is no filter motor on the telescope at all. **IHASLIM** is set to 1 if the hardware includes limit switches, else 0 if travel limits are specified entirely in software via the *home.cfg* config file. **IPOSSIDE** is set to 1 if the home switch is to be approached while moving the filter wheel in the canonical positive direction (that is, clockwise when looking through the camera), 0 if to be approached while moving in the negative direction. It is important to choose the side of the home switch that is most accurate. **IHOMELOW** allows selecting the polarity of the home switch. A value of 1 means the home voltage goes low when active, 0 means the voltage goes high when active. This also allows using either a normally-open or normally-closed home switch. **ISTEP** is the number of motor steps per full revolution of the filter wheel. **ISIGN** is set to 1 if a raw positive motor step command rotates the filter wheel clockwise if viewed through the camera, set to 0 if counterclockwise. **I1STEP** is the number of motor steps between each successive filter position. **IOFFSET** is the initial number of steps to move from home to the first filter position. **ILIMMARG** is the angular amount by which the software will avoid the hard limits, if any. This is to allow for limit switches which do not repeat precisely. **IMAXVEL** is the maximum velocity to use while moving to a new filter position. **IMAXACC** is the maximum acceleration to use while moving the filter. **ISLIMACC** is the acceleration to use when a limit switch is encountered or other severe condition is detected. See also the discussion for HSLIMACC.

The next section contains details about each filter, and default values to use when and if any details are missing or have not yet been established. The file should be edited manually when the system is first installed, then will be appended to (updated) by *xobs* automatically during he automatic focus procedure.

The file describes **NFILT** filters. Each filter requires one line, whose parameter name is **FILT*n***, where ***n*** ranges from 0 up to (but not including) **NFILT**. The format of the value portion of each filter entry consists of up to five fields and must be surrounded by apostrophes ('). Each subfield is separated by a comma (,). The following table summaries these fields:

Description	**Units**	**Default**	**Required**
Name, must begin with a unique letter, case is ignored	(none)	(none)	✔
Flat calibration time	seconds	FLATDURDEF	
Flat light source intensity	1..3	FLATLTEDEF	
Focus/temperature, pair 1	microns/°C	NOMPOSDEF/FILTTDEF	
Focus/temperature, pair 2	microns/°C	focus pair 1	

Names must be unique in the first character because this letter is used to form flat calibration file names and several places throughout OCAAS only save or report the first letter of the name. The microns focus distance referred to here is the distance from the home in the canonical direction (positive towards camera). The defaults referred to are as follows. **FLATDURDEF** is the default duration for a flat. **FLATLTEDEF** is the default light intensity to use during a flat. This value ranges from 1, dimmest, to MAXFLINT, the brightest. **NOMPOSDEF** is the nominal focus position lacking any other information. microns from home positive towards camera. **FILTTDEF** is the default temperature to assume when no other information is available, in degrees C. Finally, the parameter **FDEFLT** names the default filter to be moved into position after the Home procedure completes from *xobs.*

2.4.7 camera.cfg

This file contains information about the camera. It is used both by the batch scheduling process *camerad*, and the real-time process *camera*.

```
GSCHUNTRAD  0.0   ! max GSC match search radius, rads. 0 for 1 attempt.

HPIXSZ      1.27  ! arc seconds/pixel horizontal @ 1:1 binning
VPIXSZ      1.27  ! arc seconds/pixel vertical @ 1:1 binning
LRFLIP      0     ! 1 to flip image cols, else 0
TBFLIP      0     ! 1 to flip image rows, else 0
RALEFT      1     ! 1 if RA increases to the left on raw image, else 0
DECUP       1     ! 1 if Dec increases going up on raw image, else 0

DEFTEMP     -20   ! default cooler target, C
CAMDIG_MAX  10    ! max time for full-frame download, secs

NBIAS       1     ! number of bias frames to average together
NTHERM      1     ! number of thermal frames to average together
THERMDUR    60    ! seconds per thermal frame
NFLAT       1     ! number of flats to average together


TELE  'Engineering Model'     ! TELESCOP FITS keyword
ORIG  'CSI'                   ! ORIGIN FITS keyword

!     For RCS Only -- Do Not Edit
!     @(#) $RCSfile: camera.cfg,v $ $Date: 1998/06/27 20:36:02 $ $Revisic
$Name:  $
```

This file contains several parameters related in some fashion to the camera or image processing.

GSCHUNTRAD is the largest distance from the nominal center of an image that will be searched in order to find a match between the stars in an image and those in the GSC catalog. Larger values allow matches to potentially be found when the nominal center values are not yet well calibrated, but also allows the hunt to search for a longer time before failing or possibly finding false matches if the image quality is poor and the stars are largely bogus. The hunt algorithm starts at the nominal center and works around in an ever larger spiral, moving by 1/3 the image size each step. Each step tests stars from the catalog in a patch matching the size of the image. Setting the value to 0 limits the algorithm to exactly one attempt and it entirely adequate once the telescope axes are well calibrated.

The next several parameters indicate the image scale and orientation. **HPIXSZ** and **VPISXSZ** are the size of each pixel in arc seconds when no binning is in effect. *The GSC pattern algorithm adapts to changing translation and rotation but does not adapt to pixel scale, so it is very important to set these values accurately for proper functioning of the GSC coordinate computation facility.* The following formula is handy in computing the pixel scale:

$$P = \frac{206s}{f}$$

where *P* is the pixel scale in arc seconds; *s* is the size of one pixel on the chip in microns; and *f* is the focal length of the optical system in mm. Once you are taking good images, use *camera* to check the accuracy of the pixel scale and recomputing GSC solutions with small changes using the Field star setup Option.

LRFLIP and **TBFLIP** are each set to 1 if each image read from the camera should have its pixels flipped left-to-right or top-to-bottom, respectively. These effect the operation of both *camerad*, the batch camera daemon, and the default Expose flip controls within *camera,* the interactive camera control GUI program. **RALEFT** and **DECUP** are used to indicate how the sky is reflected within the telescope optical path and ends up on the camera CCD. These values are with respect to the actual chip, that is, before LRFLIP and/or TBFLIP take effect. No allowance is available for image rotation; this is better handled by proper physical camera mounting orientation, or field rotator calibration if applicable.

See the discussion in the chapter on *Camera* for an interactive method of determining these values.

DEFTEMP is the default temperature at which the camera cooler will be set by the system. This applies both to batch operation and serves as the default setting in the *camera* Expose dialog. **CAMDIG_MAX** serves as a timeout for the camera and is the longest time during which any image should take to be read. This should be set the time of a full frame image with no binning, and will be scaled internally as appropriate.

The next set of parameters specify the number of each type of image calibration frames to take and average when acquired via the batch operation mode. These values only apply to calibration images taken during batch mode. Arbitrary values may be set on the fly when taking calibration images interactively using *camera.* **NBIAS** is the number of images which will be taken and averaged when a bias frame is created in batch mode. **NTHERM** is the number of images taken and averaged when a thermal frame is created. **THERMDUR** is the duration of each thermal. **NFLAT** is the number of flats taken and averaged.

The value of **TELE** will be copied into the TELESCOP FITS field of each image taken. According to the FITS reference documentation NOST 100-1.1, dated September 29, 1995, this field ‘‘shall contain a character string identifying the telescope used to acquire the data contained in the array.’’ Similarly, **ORIG** will be copied into the ORIGIN field, which ‘‘shall contain a character string identifying the organization creating the FITS file.’’ OCAAS also sets the INSTRUME field using data supplied directly by the camera driver installed.

2.4.8 telsched.cfg

This file contains many loosely related parameters, most of which are used by the batch schedule preparation tool, *telsched*, or the batch execution daemon, *telrun.* Follows is an example and some explanation.

```
! CSI development platform telsched.cfg

! defaults for schedule requests
COMPRESS = 1            ! default copression factor
DEFBIN =    1           ! default camera binning, pixels
DEFIMW =    768         ! default camera width, pixels (before binning)
DEFIMH =    512         ! default camera height, pixels (before binning)
LSTDELTADEF = 30;       ! default for LSTDELTA keyword, mins

! Scheduling and equipment constraints
MINALT = 0.08           ! minimum allowable altitude for imaging, rads
MAXALT = 1.57079        ! max altitude, rads (generally only useful alt-a
MAXHA = 6.28            ! max abs ha limit for imaging, rads
MAXDEC = 1.5707         ! max +dec for imaging rads
SUNDOWN = .2094395      ! rads sun is below horizon we consider dark
SETUP_TO = 300          ! max secs to wait for devices to set up before a
IGSUN = 1               ! 1 to ignore whether the sun is up, else 0
STOWALT = 0.0           ! stow altitude, rads
STOWAZ = 3.1415926      ! stow azimuth, rads E of N
SERVICEALT = 1.57       ! service altitude, rads
SERVICEAZ = 0           ! service azimuth, rads E of N
BANNER = 'Clear Sky Institute'

! Settings for the all-sky pointing mesh
MESHEXPTIME 5           ! mesh exposure time, secs
MESHFILTER  C           ! mesh filter code
MESHCOMP    100         ! mesh images compression scale factor
PTGRAD      .1745       ! pointing mesh interpolation radius, rads

! Settings for the photometric scans
PHOTBDUR        20      ! duration for blue filter
PHOTVDUR        20      ! duration for visible (yellow) filter
PHOTRDUR        20      ! duration for red filter
PHOTIDUR        20      ! duration for IR filter

! how the progress beeps are to sound
OffTargPitch = 1500     ! bell pitch while off target
OffTargDuration   = 100 ! bell duration while off target, ms
OffTargPercent = 10     ! bell volume (%) while off target
OnTargPitch = 2000      ! same, when acquire target
OnTargDuration = 200    ! same, when acquire target
OnTargPercent = 20      ! same, when acquire target
BeepPeriod = 500        ! overall interval, milliseconds

! Local conditions -- updated dynamically is have gpsd/wxd installed
LONGITUDE   1.5975      ! site longitude, +W rads
LATITUDE    0.7271      ! site latitude, +N rads
ELEVATION   200         ! elevation above sea level, m
TEMPERATURE 10          ! air temperature, degrees C
PRESSURE    1010        ! air pressure, mB

! For RCS Only -- Do Not Edit
! @(#) $RCSfile: telsched.cfg,v $ $Date: 1998/08/07 23:11:49 $ $Revision:
$Name:  $
```

The first several parameters supply defaults for *telsched* when creating new scans from requests which do not specifically indicate these settings. **COMPRESS** is 0 for no compression, 1 for maximum lossless compression, and larger values indicate increased compression at the cost of some image data. Values up to 100 or so are still useful for photometric and astrometric analysis but the background starts to get too blocky for pretty pictures. The value is actually the *scale* parameter to the STScI H-compress algorithm which is used throughout OCAAS for image compression. **DEFBIN** is the default camera pixel binning. **DEFIMW** and **DEFIMH** are the default image width and height, in pixels. (These values are not retrieved from the camera since the *telsched* program is intended to operable off-line from any observatory hardware.)

LSTDELTADEF is the default amount of allowable difference between the time a request file specifies a starting time for a scan and when the schedular will assign it a time in the final schedule.

The next several parameters restrain *telsched* from using certain portions of the sky for scheduling observation. **MINALT** sets a minimum altitude to account for local horizon, a less than ideal dome shutter opening, or desired atmospheric seeing. **MAXALT** sets a maximum altitude and is intended to avoid scheduling in the zenith hole of an alt-az mount**.. MAXHA** and **MAXDEC** are just for problematic mounts which suffer from mechanical, cabling or other unfortunate circumstances.

SUNDOWN specifies the number of degrees below the horizon the sun must be for *telsched* to consider the night suitable for observations. 18° down is typically considered ‘‘astronomical twilight’’, as dark as it ever gets. **SETUP_TO** is the longest time to wait for all equipment to set up for an observation during batch operation. **MAXLAG** is the further behind batch operation is allowed to run an observation, later than which the observation is marked as failed and skipped. **IGSUN** can be set to 1 to allow batch operations to be performed even when the sun is above SUNDOWN; if set to 0, all batch operations are stopped and the roof is closed at dawn. **STOWALT** and **STOWAZ** set the desired position of the telescope when it is stowed at dawn, or when manually commanded to ‘‘Stow’’ from *xobs.* **SERVICEALT** and **SERVICEAZ** set the position of the telescope when manually commanded to go to the ‘‘Service’’ position in *xobs.* **BANNER** is an arbitrary string which will be used as a title in many GUI tools, including *telsched, shm* and *xobs.*

MESHEXPTIME is the duration of each image taken as part of the all-sky pointing mesh taken with *telsched*. **MESHFILTER** is the filter which will be used for the mesh. **MESHCOMP** is the compression factor to be used for each image. **PTGRAD** is the radius about the target position which is searched for pointing mesh data points when the mesh is put to use. The mesh values are weighted to have an influence of 1 if they coincide exactly with the target position up to a value of 0 if they are this far or farther. This value should be somewhat larger than the actual mesh spacing.

The next section regards taking standard Landolt photometric observations with *telsched*. **PHOTBDUR** is the default duration for taking images through the B filter. Similarly, **PHOTVDUR**, **PHOTRDUR** and **PHOTIDUR** are the defaults durations for the V, R and I filters, respectively.

The next several parameters control the pitch, duration and volume of the beeps generated by *xobs* when manual operations are underway and when they complete. These beeps are only generated when enabled.

The next several parameters specify default location and fake weather conditions. These values are always used by *telsched* when creating a new schedule since it is designed to be operable off-line away from any special equipment. Also, these value are used by the real-time systems within OCAAS if and only if a GPS receiver or weather station are not attached.

2.4.9 wx.cfg

This file defines the extreme values of wind speed, temperature and humidity above which a weather alert is asserted. During scheduled observing, a weather alert will cause observing to cease and the dome to close. Scheduled observing will resume when the alert is gone for a while. This file also contains the amounts by which each datum must change to trigger a new entry in the weather log file, $TELHOME/archive/logs/wx.log. The weather data are sampled once per second so this strategy of waiting for a value to change results in much smaller files at no loss of accuracy. Of course, these features require a functioning weather station and the *wxd* weather daemon to be running.

Follows is a typical example.

```
! CSI

! wxd config file

HAVEWX       0     ! 1 if even have a weather station, else 0

! weather alert thresholds
MINT         -30   ! min operating temp, C
MAXT         45    ! max operating temp, C
MAXH         95    ! max operating humidity, %
MAXWS        50    ! max operating wind speed, kph
ALRTTM       30    ! alert remains until all ok for this many minutes

DELWS        5     ! delta wind speed to trigger log, kph
DELWD        22    ! delta wind direction to trigger log, degrees
DELH         5     ! delta humidity to trigger log, percent
DELR         1     ! delta rain to trigger log, mm (always log when rain s
DELRT        30    ! rain alert remains after rain stops this many minutes
DELT         2     ! delta temp to trigger log, C
DELP         2     ! delta pressure to trigger log, mB

WXTTY        '/dev/ttyS2'      ! tty to weather station

! following are used to configure up to 3 DigiTemp sensors.
HAVEAUX      0     ! 1 if even have the sensor package, else 0
AUXTTY       '/dev/ttyE03'i    ! serial port where aux temp sensors are co

! For RCS Only -- Do Not Edit
! @(#) $RCSfile: wx.cfg,v $ $Date: 1998/07/09 18:01:39 $ $Revision: 1.2 $
```

HAVEWX is set to 1 if a weather station is indeed attached, else 0. If this field is 0 the weather deamon process, *wxd*, will not execute. The various **MIN?** and **MAX?** parameters specify the limit values beyond which the condition will trigger a weather alert. In addition, *any* change in the rain gauge also causes an alert. All values must return to within their boundaries for **ALRTTM** minutes, including no more rain, before the alert will be cancelled.

The various **DEL?** parameters are used for data logging. *Wxd* only logs when a parameter changes by more than its specified amount.

WXTTY is the serial port to which the weather station is connected.

Up to three additional temperature sensors may also be connected to an OCAAS computer. If this feature is desired, set **HAVEAUX** to 1, else 0. The parameters **AUX[012]** should be set to indicate how many sensors are attached to the chain. The first of these sensors defined is used for temperature compensated focusing. **AUXTTY** specifies the serial port to which the serial chain is connected. Contact CSI for more information about these sensors.

2.4.10 home.cfg

This file captures the values of the telescope axis orientation in space, and also the extreme limits of travel for each.

Fill in the rough initial values for the telescope orientation and limit switch positions once by hand. Then use the Find Limits and Calib Axes tools within *xobs* to compute better values. The new values will be appended to this file when deemed satisfactory. Since the last entry for a given name is all that counts in any config file the new values thereby override the initial values.

```
! For RCS Only -- Do Not Edit
! @(#) $RCSfile: home.cfg,v $ $Date: 1998/06/27 20:57:54 $ $Revision: 1.3
$

! CSI


! Fill in the rough initial values for the telescope orientation, limit
! switch positions and directions, and motor steps per rev once. They are
! updated as we learn better values.

! "H" refers to the longitudinal axis, ie, ha or az.
! "D" refers to the latitudinal axis, ie, dec or alt.
! "R" refers to the field rotator -- must be present but ignored if don't
! "O" refers to focus motor -- must be present but ignored if don't have.
! "I" refers to filter wheel -- must be present but ignored if don't have

! initial guesses -- fill these in once by hand for a new installation

HT          -3.         ! HA of scope pole, rads
DT          .8          ! Dec of scope pole, rads
XP          5.0         ! angle from home to beneath celestial pole,
                        !   rads ccw as seen from tel pole
YC          1.5         ! angle from scope's equator to home, rads +N
NP          0.0         ! nonperpendicularity of axes, rads
R0          0.0         ! field rotator offset, if applicable

HPOSLIM     2.0         ! angle from home to positive limit, rads
HNEGLIM     -1.0        ! angle from home to positive limit, rads
DPOSLIM     3.0         ! angle from home to positive limit, rads
DNEGLIM     -3.0        ! angle from home to negative limit, rads
RNEGLIM     0           ! angle from home to negative limit, rads
RPOSLIM     0           ! angle from home to positive limit, rads
ONEGLIM     0           ! angle from home to negative limit, rads
OPOSLIM     366         ! angle from home to positive limit, rads
INEGLIM     0           ! angle from home to negative limit, rads
IPOSLIM     0           ! angle from home to positive limit, rads

HSTEP       5760000     ! motor usteps/rev
HSIGN       1           ! cnts [1=incr -1=decr] ccw looking down from N
DSTEP       5760000     ! motor usteps/rev
DSIGN       1           ! steps [1=inc -1=dec] with lat

! end of initial manual guesses -- following are added automatically
```

The first set of parameters defines the telescope orientation. See the introduction to this section for a description of the coordinate system.

The next section defines the travel either side of home to each limit switch. If an axis does not have limit switches, these values are never updated and must be set carefully by hand. To do so, use *xobs* in conjunction with the engineering status display program, *shm*, and move each axis to its extreme travel limits and note the angle and edit here.

The next section is an initial estimate of the number of motor steps which result in one complete revolution of the two principle telescope axes. Also indicated is whether the hardware steps increase in the same sense as the canonical coordinate system used throughout OCAAS. These values will be automatically refined and updated when the Fine Limits procedure is performed in *xobs.*

If you find any additional sections in your current copy of home.cfg, these were added by *xobs*.

2.4.11 gpsd.cfg

This file captures a few basic items about the GPS receiver.

```
! CSI

! config file for gpsd.

HAVEGPS     1          ! 1 if even have a gps, else 0

GPSTTY   "/dev/ttyS1"   ! tty for receiver

! For RCS Only -- Do Not Edit
! @(#) $RCSfile: gpsd.cfg,v $ $Date: 1998/07/09 18:07:16 $ $Revision: 1.2
$
```

HAVEGPS is set to 1 if a GPS receiver is connected, else to 0. The GPS daemon process, *gpsd*, will not execute if this parameter is 0. **GPSTTY** is set to the serial port to which the receiver is connected.

2.4.12 boot.cfg

When a system running OCAAS is booted, several functions may be performed, including:

- load driver modules
- start various daemons such as *telescoped*, *camerad*, *gpsd* and *wxd*
- find the telescope axis home positions
- start batch scheduled operation
- start the OCAAS GUI interface

Various sites will have different hardware and operational requirements which will dictate the combinations and parameters for these choices. Sites might also have unanticipated booting activities which must also be captured and addressed.

These capabilities are all performed with the OCAAS *boot* script. This script is usually run from /etc/rc.d/rc.local (or from an entry in /etc/rc.d/rc3.d linked to init.d for those wishing to use a SysV startup facility). Follows is an example of how the script is typically executed.

```
# Start OCAAS
export TELHOME=/usr/local/telescope
$TELHOME/bin/boot | 2>&1 tee -a $TELHOME/archive/logs/boot.log
```

Run in this way, all output generated by this script is sent to the console for immediate inspection, and is also captured in the boot.log file for the record.

When the boot script starts executing, it reads a configuration script which captures the specific requirements for a given installation. This script uses variables to define details of which modules to load and their parameters, which daemons to start and their arguments, and other options. This script is in $TELHOME/archive/config/boot.cfg. Follows is a typical example.

```
#!/bin/csh -f
# CSI
# sourced and used by bin/boot

# set to operator login (required)
set telUser = ocaas

# set to .ini filename and desired impact level to install apogee camera
set telApogee = "$TELHOME/archive/config/ap1.ini 3"

# set to each insmod command line to run
set telModules = (                    \
    "pc39.o pc39_io_base=0x2a0"       \
)

# set to each daemon to run and any args it needs
set telDaemons = (                    \
    "telescoped"                      \
    "camerad"                         \
    "gpsd -fsad"                      \
    "wxd -ls archive/logs/wx.log"     \

)

# set to max timeout, in seconds, to home all axes
# set telHomeTel = 100

# set if want to start batch processing when boot
# set telStartTelrun

# set if want to start the GUI automatically when boot
# set telStartGUI

# For RCS Only -- Do Not Edit
# @(#) $RCSfile: boot.cfg,v $ $Date: 1998/06/27 20:36:02 $ $Revision: 1.2
$
```

In order for the OCAAS GUI tools to be started automatically with this technique, it is assumed there is a login already established on the system for the user named by the **telUser** variable in boot.cfg. The default behavior of the OCAAS installation procedure is to create this login with the name *ocaas* in its own group, also *ocaas.* This procedure made the necessary changes to /etc/passwd and /etc/group, and a home directory was made at /home/ocaas with suitable .cshrc, .login and .xinitrc files for this to work properly.

The next entry only applies if your site operates a CCD camera from Apogee Instruments, Inc. If so and you want the driver installed automatically on each boot, then set **telApogee** to *one string* consisting of the full path to the .ini file to be used to define the camera configuration, a space, then the impact parameter for the driver (see the apogee driver discussion for an explanation of impact). The .ini file referred to is the one supplied by Apogee with the camera. It is convenient for it to reside in $TELHOME/archive/config along with all other site-specific files.

The next entry is an array called **telDaemons**. This is used to indicate which daemons are to be started at boot time, and what if any arguments they should be passed. These processes must all reside in $TELHOME/bin. They will each be executed as the user specified in **telUser** except for *gpsd* which will be run as root in order to be able to set the system time from the GPS receiver.

The next entries allow specifying additional startup actions. If set, **telHomeTel** causes the boot script to home all axes of the telescope, filter wheel, focus motor and roof, depending on what is installed. The value of the variable is the maximum number of seconds to wait for all homes to be found. If set, **telStartTelrun** will start the *telrun* batch scheduling process. Whether telrun actually does anything depends on what is queued to run in the $TELHOME/archive/telrun/telrun.sls batch scan list file, but at least this insures it is standing by for directions. If this is not set here, the only way *telrun* can be started is via the Batch command in *xobs*. If set, **telStartGUI** will cause the console to login as the user named in **telUser**, which in turn will start up programs as per its .login and .xinitrc scripts. As delivered, these scripts start the X Windows server and several OCAAS and standard X programs. *Note that starting the GUI automatically this way on each boot allows anyone to operate the observatory without having to log in and hence should be considered in the context of site security.*

In addition to interpreting this boot.cfg file, the boot script also checks for two additional script files. One is $TELHOME/archive/boot.pre which, if it exists and is executable, is run before boot.cfg is processed. The other is $TELHOME/archive/boot.post which, if it exists and is executable, is run after boot.cfg is processed. If either of these scripts exits with other than status zero, the OCAAS booting process is aborted.

2.5 Setting time and location

Before proceeding, make sure the computer clock is correct and the geographical position is correct, as indicated in the engineering status display, *shm*. If the GPS option is properly installed these will always be correct. If the *gpsd* is not yet started, test it temporarily by running it from an *xterm* as follows:

```
gpsd -iv
```

This should display something, even if the receiver does not currently have a lock. Once lock is established, time and location information will be displayed. Kill the program with control-c. Any GPS receiver can be used with OCAAS as long it reports the NMEA GPRMC data sentence. We can report good experience with the small Garmin model 35 engine.

If you do not have the GPS option but have an Internet connection, you may use the program *udp_time_update* to connect to the NIST (National Institute of Standards and Technology) datagram time service and set the computer clock. You must do this from an *xterm* window. Or, again from an *xterm* window, use the *date* command to type the date and time and press Enter when it is correct. You can get an accurate voice time from the US Naval Observatory Master Clock by calling 1-900-410-TIME or tune a shortwave receiver to 10 MHz and listen to radio station WWV. Once the computer clock is correct in software the hardware CMOS clock should also be set. Do this by becoming root in an *xterm* window and typing:

```
/sbin/hwclock -wu
```

This assumes your hardware clock is set to UTC, which in our experience is the preferred method for Linux. If you must use local time in the hardware clock (because you dual-boot to Windows), leave off the *u* option.

As for latitude and longitude, you can work out your location from a topographic map for your area available from the US Geological Survey at most public libraries. Edit the values into the $TELHOME/archive/config/telescoped.cfg config file and press the *Reload* option of *xobs.*

2.6 Telescope Alignment Procedure

At a new installation, or following any activity which may have disturbed the telescope, the telescope axes must be calibrated with respect to the celestial coordinate system. Use the following procedure to calibrate the telescope axes anytime its orientation is in doubt. The *Basic* alignment procedure calibrates the telescope as well as possible assuming ideal mechanical operation. This procedure is often adequate in practice with a high quality mount. The *Fine* alignment procedure accounts for all systematic imperfections by creating and installing an error map covering the entire sky which is then used to interpolate a correction during all subsequent slewing and tracking operations.

2.6.1 Basic alignment

This section describes initial and basic alignment of the telescope axes. If you have not already done so, log in as user *ocaas* to start up the OCAAS operational environment.

2.6.1.1 Find Homes

Using *xobs*, click the *Find Homes* option in the *Controls* region. This brings up a dialog listing each axis or dome which uses a home switch for a reference. Each axis may be homed individually, or all at once as desired. For each axis specified, this procedure will cause OCAAS to move the axis and find the location of the home switch. This will establish the origin of the encoder coordinate system on each axis. This procedure must be activated each time the OCAAS host computer power is turned on. (OCAAS can be set up to perform this procedure automatically each time the power is turned on; see the example *boot* initialization file.)

2.6.1.2 Find Limits

Once the home positions are known, then click the *Find Limits* option in the *Controls* region. This brings up a dialog listing each axis which uses limit switches to safeguard excess axis travel. Limits for each axis may be found individually, or all at once as desired. For each axis specified, this procedure will cause OCAAS to explore the maximum range of motion. The angular displacement from the home position of each limit position, minus a small safety margin as defined by the **?LIMMARG** parameters in the telescoped.cfg config file, will be recorded in $TELHOME/archive/config/home.cfg. The parameters are named **?POSLIM** and **?NEGLIM** for the limit in the canonical positive and negative direction. This procedure need only be repeated if the limit switches are physically moved on the telescope.

Also during this process, the scale of the motor drive mechanism is determined. The total steps per revolution and the sign of the motors with respect to the canonical coordinate system will be stored in the home.cfg file. These values appear as the appropriate **?STEP** and **?SIGN** parameters in the home.cfg file.

The separation specified between by the POSLIM and NEGLIM values are used to compute a timeout for all axis commands. If you experience timeouts trying to perform motions, it is likely these values have somehow been set incorrectly.

For axes which do not use real limit switches, the home.cfg file should be edited to indicate to the software the maximum limit to be commanded. Do this using the Paddle in *xobs* and watching the engineering display to discover the value at each desired extreme. Then edit these into the home.cfg config file. Note that the file expects all limits to be expressed as radians from the home position, sign as per the canonical coordinate system. Also, for axis without limits, the motor step size and sign must be edited manually.

2.6.1.3 Calibrate Axes with two stars

Once the home and limit positions are known and the time and location are set, click the *Calib axes* option in the *Control* region of *xobs*. This will bring up a dialog box which contains step-by-step instructions for using up to eight stars to establish the celestial orientation of the telescope, and measure any nonperpendicularity in the mount between the polar and declination axes (or altitude and azimuth axes). This dialog is pictured and briefly described in the *xobs* section of this manual.

This step requires at least two stars be chosen and the telescope maneuvered so they are each well centered in a finder scope (or eyepiece). Choose stars which are widely separated in both altitude and azimuth and bright enough to be easily seen. Avoid stars near the celestial pole, or the zenith if the telescope is an alt-az mount.

Use the *Sky View* from *XEphem* to select each choice*.* There, the option *Enable Telescope Control* should be turned on in the *Telescope* menu. Then, using the mouse, position the cursor over the first star to be used; press the right mouse button to bring up the popup menu and select *Set telescope.* You will see that the name of your star choice appears in the first position in the *Calibrate axes* list.

Now this star must be centered in the finder. To move the telescope, click to bring up the software *Paddle* available from the *Controls* region of *xobs.* Use the Coarse controls to move at the slewing velocity. When close, use the Fine contols to move slowly. The Fine slewing rate is that specified with the **CGUIDEVEL** paramater in the telescoped.cfg config file. When the star is centered in the finder, click Mark. The indicator will turn from gray to green to show the position of the star and the current encoder values have been recorded.

Repeat these steps to select, center and mark one additional star so that two green lights are on. At any time, the Undo button will move backwards through the select and mark procedure to allow changing or redoing any of these actions.

When two stars have been marked, press Solve. OCAAS will use the information gathered about the known celestial positions of the stars and the time and encoder values of each observation to compute several quantities. These characterize the orientation and scale of the axis encoders and their relationship to the home switches on each. The resulting values are shown near the bottom of the dialog. See earlier in this section for a description of this coordinate system.

If the solution seems reasonable with regards what one would expect from the basic physical situation of the telescope, then these values may be saved in home.cfg and put into effect by pressing the Install button. Watch the display of the current telescope position near the top of *xobs.* If the solution is indeed reasonable, the scope really should be pointing where it says. Use the Paddle to verify the coordinates remain reasonable all over the sky. If they do not seem reasonable, repeat the procedure by selecting *Restart* at the bottom of the dialog, or Undoing as necessary. The next time through choose different stars.

2.6.1.4 Improve Calibration with more stars

The first time through we use just two stars to get comfortable with the process. But more stars may be used to find an improved solution. You may start over by pressing Reset at this time or add additional stars for a better solution, depending on your confidence so far.

When using more than two stars, the solution is over-determined. The computation algorithm is iterative so it is occasionally possible for the solution to be entirely nonsensical. It is important to inspect the values presented to make sure they are reasonable. The computations begin with the initial values stored in the home.cfg file so if the solution is not found or the result is poor, rethink these values, edit the home.cfg file, Reload and Reset to begin a new set of stars.

When a model has been computed using Solve, the residuals for each star are shown to their left. This is the angular separation in arc seconds between what the model predicts and where the star actually was when Marked. The stars are sorted in increasing order of residual, that is, the poorest fitting star is at the bottom of the list. If one residual is much worse than all the others it usually means the star was misidentified. Just Undo its Mark and repeat. If all the residuals are particularly large, that is, larger than what should be the expected error given the finder field of view and centering effort, it can mean the telescope is not operating properly. Check for unusual amounts of gravitational flexure, loose mounting or OTA brackets, drive or encoder slippage, or intermittent cable connections.

The final test of a new pointing model is using the finder or eyepiece to check more stars. Close the *Calibrate Axes* and choose a star from the *Sky View* just as before. But now that the calibrate dialog is gone, choosing a star will actually command the telescope to acquire and track the object. Do so now with several stars and insure they are properly positioned.

Once a new solution has been installed that works well this completes the initial phase of the Basic calibration procedure. The telescope should now be capable of centering objects in the finder and tracking them for short periods of time without appreciable drift.

2.6.2 Fine-alignment using an All-sky pointing mesh

The idea of this procedure is to take a large collection of images which cover the entire sky. Each image is solved for its WCS solution against the Hubble GSC to form an error mesh. The mesh is then used to refine pointing information from the telescope encoders during all slewing and tracking.

Before performing a Fine-alignment procedure, be sure the Basic alignment procedure has been completed and is performing well.

To perform a fine alignment, start by using *telsched*. Choose the option *Add pointing mesh* from the *Options* menu. This will bring up a small dialog box. Enter the space between each mesh point. Values between 5 to 15 degrees are recommended. Finer meshes require fewer images and hence less time but do not map as subtle variations in mount behavior. Include these images in a schedule run in the usual fashion and proceed with running the schedule as described under the Batch control description of *xobs*.

The next day, the images should be waiting in $TELHOME/archive/pointmesh. Their names will all begin with *ptg*. Inspect a representative sample of the images using *Camera* to confirm they have good WCS solutions. Then, from an *xterm* window, run the *pterrors* program on all the files.

This will generate a pointing errors map file. Name this file telescoped.mesh located in $TELHOME/archive/config. An example session might go as follows:

```
% cd $TELHOME/archive/pointmesh
% pterrors ptg*.fts > $TELHOME/archive/config/telescoped.mesh
```

Install the new mesh file by clicking *Reload* in the *Controls* region. The mesh will now be used automatically for all subsequent telescope operation. This completes the fine alignment procedure.

The format of the telescoped.mesh file is four columns. The first two columns are HA and Dec, in decimal hours and degrees. The second two columns are errors in HA and Dec, in arcminutes; that is, where the scope was commanded to point minus where it actually pointed.

It is very instructive to display these errors using a plotting program such as *gnuplot.* Such plots can reveal mechanical issues which might be addressed to improve pointing. As an example of displaying the errors in Hour angle as a 3d surface over Hour Angle *versus* Declination, use an *xterm* to start gnuplot and proceed as follows:

```
gnuplot> set parametric
gnuplot> set dgrid3d 30,30,3
gnuplot> set data style linespoints
gnuplot> set contour
gnuplot> splot "telescoped.mesh" using 1:2:3
```

To display the errors in Declination, use 1:2:4 instead. Type help for more information within *gnuplot*..

2.7 Focusing the camera

Once the telescope alignment is adequate to track a field for a few minutes, OCAAS makes it easy to focus the CCD camera. Start tracking a field containing many fairly dim stars; avoid fields with diffuse objects such as galaxies and nebulæ or which are dominated by one very bright star. Use *camera* to take a few images and the *Paddle* within *xobs* to rough-in the focus. This initial focus position need only be good enough so that the brightest individual stars can be distinguished; they can still be “baseball sized”. (Incidently, out of focus images can be very useful for investigating collimation, but that is topic beyond this manual).

Now press *Auto focus* from *xobs.* A dialog will appear which displays the current focuser position, temperature, and exposure duration. Set the exposure duration to match those with which you were seeing modest stars during the initial manual focus effort. The initial duration in the dialog is that specified by the **OEXPTIM** parameter in focus.cfg. Now press *Start.* This will cause OCAAS to automatically take several images to explore and find the best focus position, then stop and report the best position it found. As it proceeds, the position and corresponding “image quality” are displayed in the bottom half of the dialog both as a table and as a graph of quality *vs.* position. By “quality” we mean the standard deviation of all pixels within the entire area of the image, except for a border of 32 pixels around the outside. The basic idea then is to find that position which maximizes the variance of the pixels. Since stellar objects create very small sharp peaks of energy, this is equivalent to finding the position which produces the sharpest start images. By using the overall area, it also avoids statistically troublesome issues such as bad pixels, cosmic rays, non-stellar objects, and so on.

The first focuser motion will always be the value of **OFIRSTSTEP** from focus.cfg. The focus search algorithm will continue making smaller and smaller steps until the step size reaches

OSTOPSTEP. When this occurs, focus is assumed to have been attained and the procedure stops; the focuser is left in the last commanded position which is not necessarily the best. The procedure also stops if at any time an image is taken in which the standard deviation is computed to be less than **OMINSTD**. This is considered an unacceptably poor image upon which to rely.

Images taken by the automatic focus procedure are saved in /tmp in case you would like to analyze them further later. You may also watch each image being taken in real time by turning on the *Auto listen* option of *camera.*

The final graph of the focus effort should be a well defined convex arc. Multiple peaks can mean the seeing changed during the effort and the results are suspect and should not be installed. A very broad peak suggests the statistics are dominated by background levels and the result should not be used.

If the procedure finishes normally and the graph looks good, press the *Install* button. This will store the focus position along with the current filter and air temperature in focus.cfg. It will also command the focuser to the new (best) focus position. One should repeat the focus procedure with each filter and at two air temperatures separated by at least 5 degrees to build up a database of focus positions for use by the automatic temperature compensated focusing facility of the batch system.

2.8 Roof setup

OCAAS supports both roll-off roofs and rotating domes. The details are captured in the *dome.cfg* configuration file. See the PC39 discussion for electrical wiring details.

Roll-off roofs are configured as if they were shutters on a dome. Set **DOMEHAVE** to 0 and **SHUTTERHAVE** to 1. If the roof has active feedback when in position, set **SHUTTERFB** to 1 and set **SHUTTERTO** to a timeout value which surely means the feedback is late and in trouble. If it has no feedback, set these to 0 and a reasonable maximum safe assumption time.

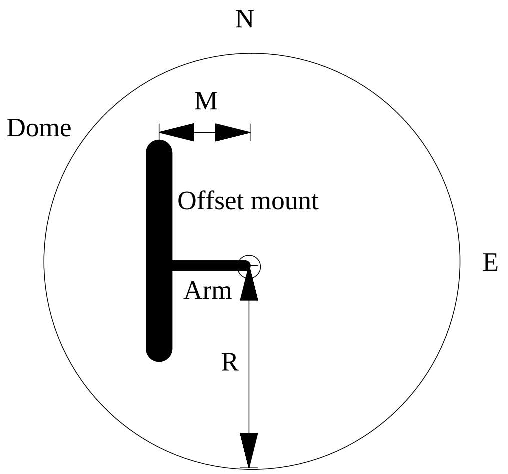
Real domes require both **HAVE** parameters set to 1. An incremental encoder and home switch are used to measure position at all times. Set **DOMEZERO** to the azimuth of the shutter opening when the dome is triggering the home switch. Set **DOMESTEP** and **DOMESIGN** according to the resolution of this encoder and whether it increments in accord with the canonical dome rotation direction (increasing values with CW motion). Most dome motors are simple on-off motors and hence can not be expected to drive to and maintain very accurate azimuth positions. Set **DOMETOL** to the allowable dome tolerance, which is generally a function of the shutter width and the rotational dome inertial load with respect to the driving motor torque. If the dome oscillates around the target position, try increasing the tolerance slightly. Set **DOMETO** to allow detecting a runaway dome.

OCAAS supports a dome which must be rotated to a fixed azimuthal position to feed power to a dome. Set **SHUTTERAZ** to the required azimuth position and **SHUTTERAZTOL** to the permissible tolerance of this position.

DOMEMOFFSET is used to allow for an offset mount, such as a German Equatorial or any mount where the optical axis is offset from the center of the dome. The assumptions made in the OCAAS implementation include:

- The telescope pole lies on or near the meridian (this includes both Alt/Az and Equatorial).

- The point at which the offset arm attaches to the telescope polar axis is at the center of the dome sphere.



DOMEMOFFSET = M/R

The value for this parameter is the ratio of the length of the mount offset arm to the radius of the dome. The sign of the value is positive if the arm is west of the mount when the telescope is pointing at the meridian; negative if it points east. For German Equatorial mounts, the value is positive if the arm is generally west of the polar axis when the telescope points to objects in the eastern sky; negative if east. See the following figure, which depicts the positive situation looking down from above.

2.9 Weather Station

OCAAS includes full software support for the Peet Bros. Ultimeter 2000 weather system. This includes measurements for air pressure, outdoor temperature and humidity, wind speed and direction, and precipitation. *Installation is the responsibility of the user.* Contact Peet Bros. directly for full information, but the following components are basically required for a complete installation: (prices are as of summer 1998 and are included for general information only)

Price, $US	Item Description
$379.00	ULTIMETER 2000 System: display, pressure, outdoor temp, wind
29.00	Heated Wind Sensor Upgrade (Ordered with System)
190.00	Heated ULTIMETER PRO Rain Gauge ($90 unheated)
110.00	Outdoor Humidity Sensor
15.00	40 Ft. Sensor Extension Cable
$723.00	Approximate system price

Once the weather equipment is installed plug in its power supply and plug the serial cable into a serial port on the OCAAS computer. Edit wx.cfg to specify the software name of this port with the

WXTTY parameter, and set **HAVEWX** to 1. Also look over the other parameters in wx.cfg and set them as desired.

Then to test for proper functioning, start the daemon manually one time from an *xterm* as follows:

```
wxd -vr
```

This should display a line of weather info once per second until killed with control-c. If nothing happens, check that the serial port is properly configured in Linux.

The only calibration required is to set the air pressure on the Peet Bros unit. Refer to Peet Bros documentation. Note that OCAAS uses the barometric pressure to compute the refraction model. This pressure should be what is called the "station pressure." This is the actual air pressure at the observatory. The air pressure reported by the National Weather Service is referred to sea level and should *not* be used. The following formula can be used to convert sea level pressure to station pressure at altitude y:

$$P = P_0 e^{-ay}$$

Where:

P = pressure at altitude y

P_0 = pressure at sea level (what the TV weather reports)

a = 0.116

y = altitude above sea level, km

2.9.1 Digitemp remote temperature sensor

OCAAS supports up to three Digitemp remote temperature sensors supplied by Nexus Computing, Inc. These can be purchased from http://www.eskimo.com/~nexus/digitemp.shtm. If you choose to use these, plug them into any available serial port and edit wx.cfg before starting wxd

If these sensors are used, the temperature reported by the first sensor will be used for weather alerts, not the weather station. All three sensor temperatures are recorded in the wx.log log and are reported by *xobs* and *shm.*

2.9.2 Starting *wxd*

To start the *wxd* weather deamon automatically each time the OCAAS computer is booted, edit boot.cfg to make sure the *wxd* weather deamon process is included in the list of daemons to start. If the entry is not already there, it should be added to the telDaemons array and look like this:

```
"wxd -ls archive/logs/wx.log"              \
```

Don't forget the \ at the end of the line to indicate continuation.

The format of the wx.log file is as follows. Each field is fixed width and all occur on each line:

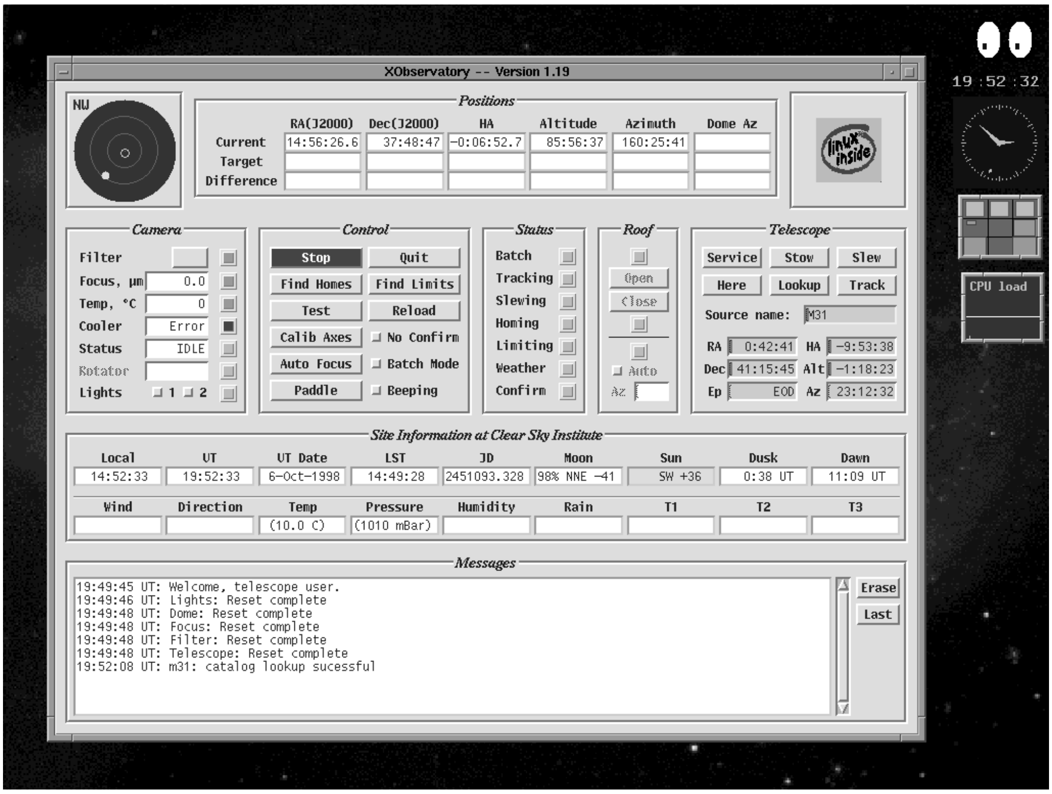
```
JD
Wind speed, kph
Wind direction, degrees E of N
Temp, C
Humidity, %
Station pressure, mbar
```

```
Rain, mm
alert code(s)
Aux0, C, or -99.99
Aux1, C, or -99.99
Aux2, C, or -99.99
```

The alert codes are T, C, H, W and R, for maximum temperature exceeded, minimum temperature, maximum humidity, maximum wind and presence of rain. If you have also installed one or more Digitemp remote temperature sensors, their current values are listed in the final three columns; the values are set to -99.99 for each sensor not installed.

3 OCAAS Desktop

This chapter discusses the overall OCAAS graphical desktop environment. The individual applications are mentioned only in passing. Rather, the focus here is on those issues which effect the usage of the system as a whole.



3.1 X Windows

OCAAS uses the X Windows system augmented with the Motif toolkit for all graphical user interfaces. This means that each application follows a consistent set of display and behavioral rules for its own controls, and also means the various applications interact with each other in a consistent manner. Since X Windows is an open system, many other utilities and applications are available which may of general use to an OCAAS environment, although they are not included with the base system.

Once such example is the ability to display all OCAAS tools remotely. All that is required is a live internet connection with the OCAAS control computer, and an X Server. These servers are available without charge for several systems, including MS Windows. Contact CSI for our current recommendations.

In the X Windows architecture, the underlying video display hardware, as well as the particular brand and connectivity of mouse and layout of the keys on the keyboard are all hidden from the individual application programs. All machine dependencies are captured in the *X Server* process. Only this process communicates directly with these devices; all X Windows client applications communicate through a socket (or possibly shared memory if running on the same system) to the

X server to learn what the user is doing and to draw on the screen. This strict separation allows for tremendous flexibility regards to running X Windows applications in a heterogeneous networked environment.

All "useful" work is performed by *client* programs. These include the ubiquitous X tools such as *xterm, clock* and such, as well as OCAAS tools such as *xobs* and *xtelescope.* One particularly important client is the *window manager* which is described shortly.

For more information about X Windows, type "man X" and see the list of references below.

3.2 Default desktop arrangement

The default OCAAS desktop is a screen with several tools arranged down the right edge of the screen. The background image of M51 is known as the *root window.* The initial root window, among these other X Windows start up activities, is set from the configuration file *.xinitrc* in the OCAAS home directory. The cursor shape is a thick **X** when over the root window.

The top right application is *xeyes.* It follows the cursor around on the screen. It is just whimsical, but can occasionally serve as an indication that the system is alive if ever in doubt.

Next down is a digital clock known as *xdaliclock.* It is configured to display the current UTC time to the nearest second. Placing the mouse over any portion of a numeral, as indicated when the cursor shape changes from the thick X of the root window to an arrow pointing to the upper left, and clicking any button will change the display to the current UTC date as Month/Day/Year until the mouse button is released.

Next down is *xclock.* It is configured to display the current local time to the nearest second as an analog clock face.

Next down is the *fvwm* window manager *Pager* control panel. This is discussed in a separate section.

Next down is *xload,* a client which displays a graph of the current and recent system *load average.* The load can help gauge the level of overall system activity. The white line indicates a load average of 1. If the average ever reaches two or more, the vertical scale of the graph will be compressed and additional lines will be added to allow displaying the graph. Linux is a multitasking operating system and so can and usually is running several different processes ostensibly at one time. The load average is the number of such processes which are eligible to run if they were given an opportunity. The higher the load average, the less CPU time each runnable process is allocated and the less work it accomplishes. This is all fine up to a point, but too many competing processes can end up causing more context switching and paging overhead than useful work. This is known as *thrashing.*

In addition to the persistent tools on the right edge, the OCAAS start-up script also starts several of OCAAS operation tools, each in its own virtual screen. These include *xobs, camera, xephem* and one general purpose command window, *xterm.*

The appearance and the initial programs started for the desktop are determined largely by the contents of the file *.xinitrc,* located in the home directory of the *ocaas* telescope operator login. The only exception is the fvwm pager, which is configured in the file *.fvwmrc,* also in ocaas's home.

3.3 Mouse

The mouse is used to move the cursor around the screen and operate the various controls of each application. The applications are discussed elsewhere. This section discusses the use of the mouse for general administrative tasks by using it with the root window.

Positioning the cursor anywhere over the root window and pressing a button will bring up a *popup menu* containing a list of selections. Keeping the mouse button depressed, move up and down and you will see each item in turn is made to appear raised to indicate it is *armed.* Releasing the mouse button over an armed item will activate it. Moving the mouse away from the popup menu so that no item is armed and releasing the button will make the popup disappear with no effects. Some of the items are just the lead-in to an another *cascading menu.* Move the cursor a little to right then move up and down to arm or activate these items. A cascading menu operates the same as the main popup menu but helps to organize long lists off the main list.

If the action results in a new program being started,. the next thing that happens is a rectangular outline will appear showing the size of the new application but the location is not determined. The cursor will be connected to the outline. Move the cursor to position the application where desired and click the left button to plant it and let it start operating. During this positioning period, the cursor may even roam off the edge of the physical screen, which is taken to mean move to the next adjacent virtual screen in that direction, if any.

3.3.1 Left button over root menu

Pressing the left mouse button while over the root window brings up a short *Utilities* popup menu.

The first item brings up a general purpose command window, *xterm.*

The second item is a cascading menu of several OCAAS applications. Many of these are started automatically by logging in as *ocaas* but they are also listed here in case they are needed for any reason. In any case, these are here for convenience only; starting them from this menu is not different than starting the program from a command line. The Shutdown and Restart options are quick ways to kill most of the OCAAS programs or restart them if this should ever be necessary.

The next item is another cascading menu for general Linux application programs. Certainly many other useful tools are available but these are a random sampling. These may include the following; consult other documentation for details of their operation:

Application	**Description**
ghostview	Postscript viewer
gnuplot	2-d and 3-d plotting program
top	system activity monitor
xcalc	scientific calculator
xedit	simple screen based text editor
xmag	magnify any portion of the screen
xv	display image files in many formats; also capture and save screen images to disk

The next item will log out the *ocaas* session and return to the black Linux login console screen. Because of the impact of selecting this accidently, the real logout requires carefully selecting a second confirmation from a small cascade menu to the right. Whether logging out also stops the telescope depends on how the system is configured.

The bottom item causes all applications to redraw themselves. This is useful in case of a bug.

3.3.2 Middle button over root menu

With the cursor positioned over the desktop, press the *middle mouse button*. This brings up a popup of basic window operations. These are the same things which can be performed to top level windows from their individual decorations. The reason this is here is for those cases when the corresponding decoration area is not visible for some reason such as off the screen or buried by another window, or when the decorations have been removed (for example, the OCAAS clock) or reduced in scope (for example, the OCAAS load graph).

To use this menu, slide the cursor over the desired operation and release. This will change the cursor to a solid circle. Now move the cursor so it is over the desired window to be effected and press the left button. Some actions take effect immediately without any further interaction required, such as Lower or Delete. Others require further interaction such as Move or Resize. In these cases, press the left button, leave it down, the cursor shape changes to a fleur until released during which time you may effect the desired change.

3.3.3 Right button over root menu

When the cursor is over the root window and the *right mouse button i*s pressed, this brings up a task list popup menu, listing the title of each X Windows application which is currently running. This is very handy to retrieve a window that has been iconified, moved off screen or buried, or just to verify whether it is running at all.

3.4 *fvwm* Window manager

Every X Windows graphical user interface delegates certain policies germane to the operation of the overall desktop to a separate specialized client application known as the *window manager.* The role of this client is to provide a means for the user to manage the very busy yet severely limited screen space. It provides the rules and methods to control such issuers as size, location, keyboard focus, and stacking order of the other applications. The window manager generally has not window of its own, but provides its services through popup menus off the root window, by decorations which it adorns other applications, and by special keyboard ‘‘hotkey’’ sequences.

OCAAS uses the popular window manager known as *fvwm*[*]*.* It is very capable and is the manger of choice on many Linux systems. It is also very configurable which can be seen either as an opportunity to customize its operation as desired, or as a complication. The author of OCAAS hopes you find the choices made here to your liking. All fvwm configuration is performed by editing the file *.fvwmrc* in the teloper home directory. Doing so should not interfere with the OCAAS applications so you are free to do so as you wish.

Follows is a brief description of the basic services provided by fvwm as configured by OCAAS.

---

[*] Later versions of OCAAS may switch to the CDE.

3.4.1 Window decorations

Around each *top level window* the *window manager* draws controls by which the window, and hence the application, may be manipulated.

3.4.1.1 Resize handles

Around the outer edge are the *resize handles* which allow the window to be moved or resized. The handles along the edges allowing moving, while the handles in the corners allow resizing. Moving past the edge of the physical screen is interpreted to mean move to the next virtual screen in that direction, if any; watch the pager display if you get lost.

3.4.1.2 Title bar

Across the top center is the *title bar.* This displays the overall name of the application. The text displayed here matches that displayed by the task list popup menu displayed by clicking the right mouse button while the cursor is over the root window. Clicking and holding the title bar with the mouse allows the window to be moved.

3.4.1.3 Maximize button

The square just inside the upper right corner is the *maximize button.* Clicking on this toggles between a ‘‘normal’’ size window and one tends to fill the entire screen. The exact behavior of this button varies by application.

3.4.1.4 Iconify dot

Just to the left of the maximize button is the *iconify dot.* Clicking this button will make the window all but disappear, leaving only a small icon on the desktop as a reminder the application is still in fact running but can not display anything. If the application has any accompanying supporting *dialog boxes* they too will disappear when the application is iconified. To return the application to the screen again, double-click on the icon.

3.4.1.5 System menu

Just inside the upper left corner is the *system menu.* This allows several basic window operations to be performed.

Move and *Resize* allow the window to be relocated or resized as desired.

Raise causes the window to move to the front of the window *stacking order.* For example, if a portion of an application’s window is obscured by another window, raising it will cause it to move to the front of the pile and become completely visible (subject to its position with respect to the edges of the screen, of course).

Lower has the opposite effect, that is, it buries the window so all other windows lie on top of it on the desktop.

Iconify has the same effect as the iconify dot and is present only for completeness.

The next item in the menu allows you to toggle whether fvwm considers this window to be *sticky.* When this is activated, the window will remain visible on all virtual screens; when off, the window is only visible while the virtual screen on which it is located is the current screen. (Virtual screens are discussed under fvwm).

Destroy is a rude way to kill the application. It should never be used unless you are desperate because it does not allow the application to perform any shutdown activity; it should only be used to kill an application which appears to have gone awry.

Delete is the appropriate way to end an application gracefully. (This button is often grayed out although it is still available; it appears to be a bug in fvwm).

The last item in the system menu is *ScrollBar.* This will place the application in a separate overall set of scroll bars which allow it to be panned. This tends to be a bit confusing but is handy for an application that for whatever reason has become manageably large and yet has not controls it provides to deal with this large size. There is no way to remove the scroll bars once applied.

3.4.2 Keyboard

The fundamental issue with regards to the keyboard is one of ambiguity: there are many different applications potentially on the screen yet there is only one keyboard. When you press a key, which application gets it?

The X Windows architecture solves this issue by use of a *focus policy.* The window manager has the authority to provide a steering paradigm which is used to “focus” the keyboard on any one application, and to establish the physical interactions required of the user to manipulate which application has the focus at any one time.

The fvwm window manager can provide several focus change policies. The one chosen for OCAAS requires the user to use the mouse and click on the border of an application’s top level to assign it the focus.

3.4.3 Virtual screens

The typical OCAAS video display is configured for 1024x768. But this is often not enough for the many applications may be run simultaneously. Fvwm provides a way to pretend you actually a much larger display area, although you can only see one 1024x768 portion of it at a time. Each such portion is referred to as a *virtual screen.*

3.4.3.1 Pager

In order to use one physical screen for many virtual screens, one at a time, fvwm provides a *pager* control. The pager control displays the complete collection of all virtual screens as a tiny schematic. The default location of the pager chosen by OCAAS is in the set of tools down the upper right of the screen. The pager is the blue box divided into 3 rows and 3 columns. Each small square bounded by the thin blank lines represents one of nine virtual screens. The virtual screen which is currently being displayed on the physical screen is shown in pink. The application with the keyboard focus is shown in red.

To change which virtual screen is actually being displayed, place the cursor in the desired small box and clock with the left mouse button. The pink square will move to the box chosen. But more dramatically, most of the applications that were displayed will disappear, and different ones will appear. This is because your physical screen is now connected to a different virtual screen. Applications generally reside in just one virtual screen. In this way, many applications may be run at once and assigned to different virtual screens, keeping the physical screen relatively uncluttered.

Another way to move virtual screens is by holding down either Ctrl key and pressing one of the four arrow keys. This will move the physical screen to the virtual screen in the corresponding direction.

The pager control shows more. Each application's top level window is shown as a tiny light blue rectangle, corresponding to its size and location within its particular virtual window. This gives you an overall picture of *all* the applications running on the entire desktop.

To move an application around within its virtual screen or to a different screen entirely, place the cursor over its small square in the pager, press the middle mouse button, drag the application as desired and let go.

To move the physical screen around among the virtual screens, place the cursor over the pager and press the right mouse button, drag the physical screen around (as indicated by the pink square) and let go. This is not unlike using the left button except that the boundaries are not enforced and the screen may be aligned anywhere. You may also move the physical screen around using the keyboard: press either Alt key, then pressing one of the arrow keys will move the physical screen in the corresponding direction by about 10% the size of the screen.

3.4.3.2 Sticky windows

Some applications may be so important or useful that they deserve to remain visible on all virtual screens. Instead of running one separate copy of the application in each window, fvwm allows the application's window to be marked *sticky.* This means it will remain up on all virtual windows. Whether an individual window is sticky may be controlled using its system menu (or the window operations available via the middle mouse button over the root window). To avoid having to do this each time, the application may also be marked as initially sticky in *.fvwmrc.*

3.5 Additional references

The following URLs are recommended sources for additional background information about these technologies and some leads to additional applications. The technically-correct *http://* prefix has been removed from each for brevity. Please contact CSI for further assistance in selecting additional supportive reference materials.

URL	**Description**
www.ora.com/catalog/linuxnut/	*Linux in a Nutshell.* Friendly but thorough reference to using many Linux commands
www.oreilly.com/catalog/runux2	*Running Linux.* Guide to installing and administering a Linux system.

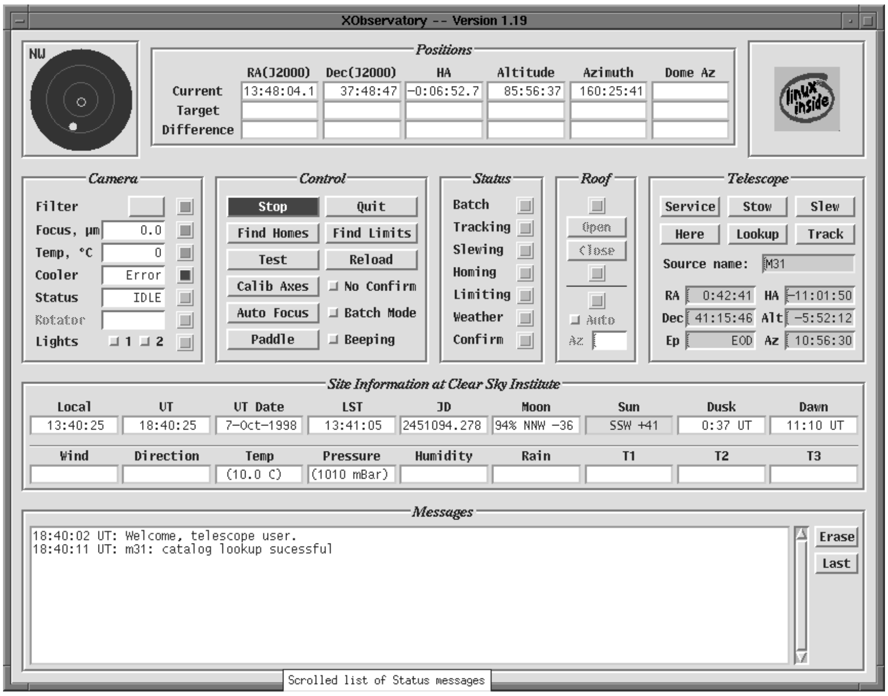
www.oreilly.com/catalog/v8	*X Window System Administrator's Guide.* Details of installing, configuring and running X Windows programs
www.sobell.com/linux.html	*A Practical Guide to Linux.* In-depth description of many Linux commands
www.xnet.com/~blatura/linapps.shtml	*Linux Applications and Utilities Page.* A general collection of many disparate Linux applications
SAL.KachinaTech.COM	*Scientific Applications on Linux.* An index to tools available for Linux specifically applicable to scientific research and data analysis
www.intercom.ru/~dima/fvwm/fvwm.html	*fvwm documentation.* FVWM is the window manager suppied by OCAAS.
www.geek-girl.com/unix.html	*UNIX Reference Desk.* An on-line collection of many general UNIX reference works.
www.camb.opengroup.org/x	*X Window System* is now managed by the Open Group

4 Real-Time Operation

This section discusses those portions of OCAAS which provide for direct control of the telescope and observatory. Direct control of the camera is largely contained in the separate section on the *camera* program.

4.1 xobs

This program is used to make direct commands to the telescope, focus, filter wheel, dome and dome shutter. This program can command several setup procedures, including finding *Home* and *Limit* switches; calibrating telescope axis orientation; performing an automated focusing procedure and reloading configuration files. This program also controls whether the unattended batch observing sequencer program, *telrun* is engaged. Direct control operations are inhibited as long as *telrun* is active.



Note: The idea of *xobs* is to allow *direct control* of the observatory. Xobs makes no attempt to decide whether a command is sensible. The only restrictions on motions are those imposed by the hardware limit switches or, if absent, corresponding maximum travel values in the configuration files. Automatic features that are part of batch operation, such as temperature compensated and filter-specific focusing, stopping when a weather alert is in progress or closing the dome at dawn are *not* performed when xobs is in control of the system. Xobs lets you do what you say. no more and no less.

This tour will work across the xobs interface by region, left to right, top to bottom. But first some topics of general interest.

Only text fields which have a gray background may be edited; white fields are for display only. The Moon, Sun, Dusk and Dawn fields are yellow when their respective topic is contrary to a dark sky.

The status indicator lights may be in one of four colors. Gray indicates no activity, not installed or irrelevant. Yellow means progress is being made towards a goal or operating near the extremes of acceptable limits. Green means functioning within limits. Red indicates trouble of some sort, such as operation not progressing as expected, operating outside of limits or safety issues may be at risk.

When operating in Batch mode, many of the controls will become impossible to manipulate. This is to prevent accidently interfering with a scheduled operation in progress. Even so, most fields still display accurate information. In this way xobs can serve as a passive status display, as well as a control station. Control may be returned to xobs if Batch mode is turned off, as described in the discussion for the Control region controls. If a second or subsequent instance of xobs is started when one is already running for the same telescope, it will also go into passive mode. In this situation, however, it will never be possible to gain control. Only the first xobs to attach to a telescope can ever control that telescope. As many instances of xobs as desired may be running at one time, but only the first can control the observatory.

As the cursor moves over the various controls and fields, a little ‘‘tool tip’’ window will appear with a brief but helpful description of the function*. They are generally helpful but can be deactivated as described next.

All of the control buttons provided by xobs which result in some mechanical or electrical action first ask for a confirmation before they take effect. This is an effective approach to safety but can become annoying to an experienced operator. These confirmation messages can be toggled off and on using the Confirmation toggle button. Also, xobs will start in the state of not asking for confirmations if executed with a -quiet argument. The only exceptions are the Stop button which never asks for a confirmation and the Exit button which always asks for a confirmation before exiting the program. Also, when confirmations are off, the tip windows are also inactive.

Many of the features of xobs utilize information stored in config files. These reside in the directory $TELHOME/archive/config unless otherwise stated.

If you would like to read through this description using a live xobs but are not on a computer connected to a real telescope, note that all xobs needs in order to run is to find the control fifos which are created by the daemon processes. This can be accomplished by telling the *telescoped* daemon to run in a ‘‘no hardware’’ configuration. Do this by running **telescoped -h** then **xobs**. The simulation is not very interesting, but it is sufficient to fool xobs.

4.1.1 Sky Dome Graphic

The dark circle in the upper left region depicts the entire visible sky as though one was looking down on the surface of a flat earth. North is up, west is left. The two circles are lines of 30° and 60° altitude; the horizon is at the outer edge of the dark circle.

* Another reason these are here is because I was challenged by an M$ Windows programmer who said it couldn’t be done in X Windows.

Four symbols may appear on the map. An open white circle denotes the current pointing direction of the telescope. A green X denotes the current target pointing position of the telescope. A filled yellow circle denotes the Sun. A gray crescent denotes the Moon.

4.1.2 Positions

In the top center region is displayed the current and target pointing directions of the telescope and the dome (if installed). Also displayed is the amount by which they differ. The Target and Difference rows are blank if the telescope is idle.

4.1.3 Logo

In the upper right region is an image of your choice, up to 100x100 pixels. Larger images will be cropped, smaller images will be centered. The format must be X pixmap, as indicated by a suffix of .xpm. The file displayed is $TELHOME/archive/config/logo.xpm. If you have an image in a different format, you might try using *xv* to convert it since it supports many file formats. When creating the .xpm file, use no more than 50 distinct colors.

4.1.4 Camera

At the left end of the center bay of regions is one which manages issues related to the CCD camera. This does not include taking actual images; for that use the *camera* program.

4.1.4.1 Filter

The top control shows the current filter, if a filter wheel is connected. The light is green if that filter is in fact in place, yellow if it is coming into position, red if something is preventing it from attaining position. To change the filter, clicking on the filter name will display an option menu. Slide to the desired filter and release. The light will turn yellow, then green when the new filter is in position. The names of the filters and their wheel positions are defined in the filter.cfg config file.

4.1.4.2 Focus

The next row displays the current focus position. The units are in microns from the home position of that control axis, increasing positive towards the main mirror. To change the position use the Paddle, described shortly. The indicator light will be yellow when the filter is moving and be green when the commanded position is reached.

4.1.4.3 Temp, ° C

The next row displays the current CCD camera temperature. The indicator light is green as long as the Cooler status (next row) indicates At Targ.

4.1.4.4 Cooler

The next row indicates the status of the CCD camera thermoelectric cooler. The possible states and the colors on the indicator are as follows:

Status	Light Color	Description
At Targ	Green	Cooler is at target temperature and holding
< Targ	Red	Cooler temp is more than 2 °C colder than target
> Targ	Red	Cooler temp is more than 2 °C warmer than target
Off	Gray	Cooler is turned off
Ramping	Yellow	Cooler is working towards target temperature
To Amb	Yellow	Cooler is doing a controlled warming to ambient
Floor	Red	Cooler is at max effort and can go no cooler
Ceiling	Red	Cooler temp can not go any higher
At Amb	Gray	Cooler has reached ambient temp and is off
Error	Red	Cooler is in an unknown state

4.1.4.5 Status

The next row indicates the status of the CCD camera electronics and computer interface. It can be in one of four states. IDLE means the camera is not currently busy; the indicator will be gray. EXPOSING means the camera is busy acquiring an image; the indicator will be yellow. READING means the computer is reading pixels from the camera; the indicator will be yellow. ERROR means something is wrong with the camera interface; the indicator will be red. Depending on the type of camera interface, system response may be poor while READING.

4.1.4.6 Field Rotator Position

The last line in the camera region applies only for an Alt-Az telescope with an image rotator. The value displayed is the rotator position angle from home, increasing positive while rotating clockwise as seen looking through the camera. The indicator will be yellow if it is moving into position; red if it is not moving and not within tolerance; green if it is in position.

4.1.5 Control

The Control region of xobs contains functions related to system configuration and setup.

4.1.5.1 Stop

Perhaps the most important is the Stop button which causes all telescope, dome and camera activity to cease. This button is intentionally red to serve as a prominent “panic button.” This button is performed immediately without prior confirmation, even when No Confirm is active.

4.1.5.2 Quit

The Quit button terminates the xobs program. This button is always first confirmed, even when No Confirm is active.

4.1.5.3 Find Homes

Find Homes brings up a dialog with which to instruct the system to start seeking the home switches on any one axis or all installed axes. Axes not installed are grayed out. Which telescope axes are installed is defined in the telescoped.cfg config file. Filter.cfg, Focus.cfg and Dome.cfg define whether these axes are installed, respectively. Also in these files are parameters which define which direction to move when taking the final home reading, i.e., which side of the home switch to use; whether the home switch is physically wired to be normally closed (low) or normally open (hi); and the direction to try first. If the starting direction is not correct, everything still works, it just takes longer because the axis will move to the opposite limit and turn around before finding its home.

Starting home on one individual axis will stop homing on all other axes.

4.1.5.4 Find Limits

Find Limits instructs the system to start moving the principle telescope motors in search of each pair of their hard limit switches. The motors move at half their usual slew speed in an attempt to be somewhat gentle. When a limit in one direction is encountered, its position is recorded and the motor reverses to find the other extreme. When the other limit is reached, its value is also recorded and the axis is moved slightly away from the limit position. The amount of this margin is set in the telescoped.cfg config file. The values are recorded as angles from the home switch position in the home.cfg config file. This procedure need only be done once during initial installation, or if the limit or home switches are moved for some reason.

If this procedure is performed when the limit switches are not functioning properly, bogus limit angles will be entered into home.cfg. The software will honor these limits and refuse to move to the real limits. The software also uses the total separation between the recorded limits to compute a time limit on all motion commands. If the stored limits are incorrect, the timeout will also be incorrect, usually shorter than expected. The solution in all cases is to edit the bogus entries out of home.cfg, Reload, fix the hardware, and find limits again.

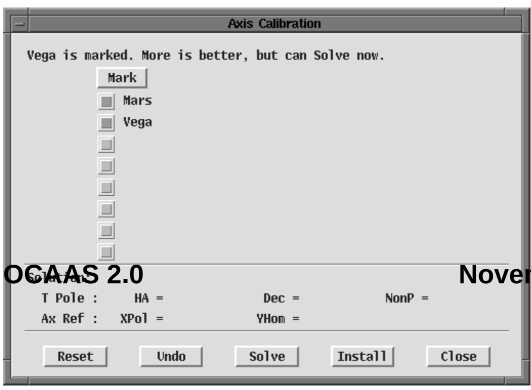
Starting limits on one individual axis will stop finding limits on all other axes.

4.1.5.5 Reload

Reload instructs most OCAAS processes to reread all config files. Do this after any one of them have been edited and you want it to take effect with the new values.

4.1.5.6 Calib Axes

This dialog is designed to help organize the steps which will calibrate the telescope axes. The procedure works well, but requires some study.



The idea is to site at least two known stars at widely separated positions with the telescope, and use this information together with geographic location, time and encoder values to establish the values of a mount model.

When only two stars are used, there is only enough information available to model an ideal two-axis mount, ignoring any non-perpendicularity of axes. With three stars, the solution includes this effect but is over-determined by only one degree of freedom. With more stars, a worth-while least-squares fit is performed. Up to the eight stars are supported by the dialog.

It is suggested that the procedure be performed at least twice. Once using just two stars to find an initial model. Inspect this module as described below and install it if it looks reasonable. If not, use different stars and repeat. Once an initial model is found, perform the setup with several more stars, preferably again all different. The initial conditions will utilize the initial model from the two-star effort and will very likely find a solution which might look roughly the same but is in fact far superior.

This dialog must be used in conjunction with the XEphem Sky View. There, set *Enable Telescope Control* in the Telescope menu. Using the right mouse button over the selected object in the Sky View, choose *Set Telescope.* The name will appear in the next row of the Axis Calibration dialog and its light will be gray. Use the Paddle in Xobs to center the object, then press **Mark**. The time, position and encoder values will be recorded and the light will turn green. Repeat for additional objects. The last Mark or Object may be removed from the list by using the **Undo** button. This may be repeated as often as desired to work back through the list. The **Solve** button uses the marked stars to compute an orientation model. If successful, it will be displayed near the bottom. (Also, the points used to compute the model are dumped for the record to the xobs.log log file.) If the new model looks reasonable, it may be put into effect by pressing the **Install** button.

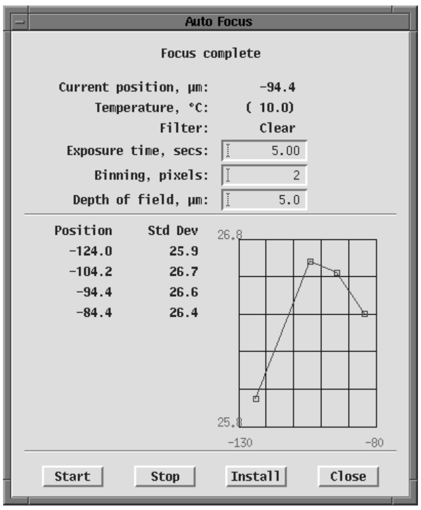
Refer to the Basic Alignment procedure in the installation portion of this manual for more information about setting up your telescope axes.

4.1.5.7 Auto Focus

This button brings up a separate dialog which largely automates focusing the CCD camera.

The idea is to take several images of the same scene at various focus positions; analyze each for quality based on the standard deviation within a central region; and keep moving to find positions which further improve the quality.

Three parameters may be adjusted before beginning. The Exposure time need only be long enough to show stars. The default is OEXPTIM in focus.cfg. The default binning is set from DEFBIN in telsched.cfg but can be set here as desired. Higher values can be helpful when first trying this feature because it runs faster, but unless your pixel scale is grossly over sampling star images with your seeing conditions use a binning of 1 when ready to do the best effort. The automatic search algorithm will stop when it wants to move the focus less than the given Depth of field. The default value is OSTOPSTEP in focus.cfg.

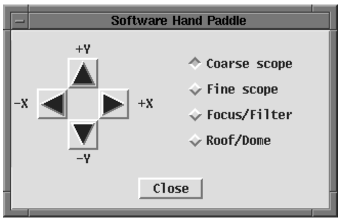


Focus.cfg also defines other parameters relevant to auto focusing which can not be adjusted from the dialog so check there before proceeding. These include the name of the filter to use, the minimum standard deviation to allow, the initial step size, and whether to require the telescope to be tracking during this procedure. Because it is only sensible to compare the standard deviations among images of the same scene, tracking is required by default and should probably remain set. But on one occasion focusing was performed using an artificial star-like object while the camera was stationary so this feature was added to allow for that gracefully.

To begin the procedure, press Start. Xobs will move the focus, take the images, do the all the analysis, and deduce the optimum focus setting automatically. As it moves the focus, the current position is displayed at the top (it is in the main xobs display too) and a table and graph of each result is maintained at the bottom section.

When a solution is found it may be Installed in focus.cfg along with the current temperature. Storing focus with temperature allows OCAAS to perform temperature compensated focusing during batch scheduled observing. It does this by linearly interpolating the current temperature against the nearest two surrounding temperatures in the config file. In order for this to work, you must use this procedure at two temperatures which differ by at least to store up the required reference values. And, of course, you must have a weather station connected.

4.1.5.8 Paddle



The software paddle is designed to be reminiscent of the traditional hardware game pad. At the left are four arrows, which serve as direction controls. At the right are four buttons which choose to which axis or motor the arrows refer. When the selection is made, the labels on the buttons change accordingly. Motion occurs while an arrow button is depressed and stops when it is released.

The top two choices connect the arrows to the telescope at either of two speeds, Coarse or Fine. When the telescope is stopped, the arrows refer to the raw telescope axes. While an arrow button is depressed the corresponding axis will move at a speed based on the MAXVEL and CGUIDEVEL parameters in telescoped.cfg. The scope will stop when the arrow is released. When the telescope is tracking, the arrows refer to HA and Dec. The speeds are based on CGUIDEVEL and FGUIDEVEL parameters. When the arrow is released, the new net position will become the target, will be reacquired and tracking will resume at the new location.

The third button connects the left and right arrows so they rotate the filter wheel and the up and down arrows move the focus out and in.

The fourth button connects the left and right arrows to rotate the dome, and the up and down arrows open and close the dome.

4.1.5.9 No Confirm

The No Confirm toggle chooses whether Xobs will first ask “are you sure” before proceeding with any command which results in motor activity. When the toggle is depressed, confirmations are not asked and the Confirm status light is red as a constant reminder. Also, when the confirmations are off, Xobs does not present the floating “tips” which appear next to the controls as they are approached by the cursor.

4.1.5.10 Batch Mode

Batch Mode toggles whether xobs is in control of the observatory or if it has delegated control to the batch scheduling daemon process, telrun. While batch mode is active, xobs disables all of its controls and becomes a passive status display to prevent accidently disturbing scheduled observations. Also while in batch mode, xobs displays a separate dialog box containing information about the current or next scheduled observation to take place.

4.1.5.11 Beeping

The Beeping button toggles whether xobs will beep rapidly while a setup action is in progress, such as slewing to a target or moving to a new filter. The duration, pitch and volume of these beeps may be set in the telsched.cfg config file.

4.1.6 Status

This region contains indicators to report a variety of general conditions.

The **Batch** light is green when telrun is operating and in charge of performing scheduled observing.

The **Tracking** light is green when the telescope is tracking an object to within TRACKACC in *telescoped.cfg*. It is yellow when it is moving into position (called ‘‘hunting’’).

The **Slewing** light is green when the telescope is moving rapidly to a target position.

The **Homing** light is green when OCAAS is searching the home switch positions on each axis.

The **Limiting** light is green when OCAAS is searching for the limit switches on the telescope axes.

The **Weather** light is green when weather statistics are within acceptable limits, red while any of these limits are exceeded, grey if no weather data is available. The limits are defined in the wx.cfg config file.

The **Confirm** light is red when the No Confirm toggle in the Control section is depressed, that is, when commands are not first confirmed.

4.1.7 Dome

The Dome region of xobs is divided into two sections. The top section is for opening or closing the shutter, or a roll-off roof. The bottom section is for controlling the azimuth of a rotating dome. Which one(s) of these is appropriate to your observatory is defined in the dome.cfg config file.

Click on the Open or Close buttons to open or close the roof. The corresponding indicator light will be yellow while the action is in progress. The light will be green when the roof is known to be in the given position. It will be gray if the position is unknown.

If you have a dome, the bottom section will allow you to rotate it to a given azimuth. Type the desired azimuth, in degrees east of north, in the text field and press Enter. The indicator light will be yellow while the dome is turning, and become green when it is in place. Press the Auto button to engage a feature where OCAAS automatically maintains azimuth lock on the dome to correspond to the current pointing direction of the telescope. While Auto mode is on, manual entry of an azimuth is not allowed and the azimuth text field becomes white to show it can not be edited.

OCAAS can also manage a dome shutter which is powered by power take-off wiper feeds which engage only at a single azimuth. In this situation, clicking on Open or Close will cause the dome to leave its current position (be it either from a manual entry or due to Auto mode), rotate to the proper azimuth to deliver power to the shutter, open or close the shutter, then return to its original azimuth and resume operation.

The dome.cfg file contains several simple parameters related to dome and shutter operation. These are discussed in the config file chapter of this manual. Be sure to set these up properly for your site before attempting to use the Dome control features in xobs.

4.1.8 Telescope

This region of xobs allows direct control of the telescope. It also provides a general purpose database lookup facility and coordinate conversion calculator.

4.1.8.1 Service

The Service button is a convenience command to drive the telescope to its service position. The location of the stow position is stored in the configuration file telsched.cfg. The altitude and azimuth of the service position are stored in the configuration parameters SERVICEALT and SERVICEAZ, respectively. Some users prefer to set these to the same values as FLATTALT and FLATTAZ (set in dome.cfg) so this button makes it convenient to manually take dome flats.

4.1.8.2 Stow

The Stow button is a convenience command to drive the telescope to its stow position. The location of the stow position is stored in the configuration file telsched.cfg. The altitude and azimuth of the stow position are stored in the configuration parameters STOWALT and STOWAZ, respectively.

4.1.8.3 Slew

Clicking on this button will drive the telescope to the position indicated in the Alt and Az fields, in the lower section of this region, unless these are beyond the current limits. If these fields are not set but others are, they will first be computed as described in the section about the calculator ability of this region.

4.1.8.4 Here

Clicking this button loads the current coordinates of the telescope into the text fields below. This feature is handy when you happen to see something interesting in the eyepiece and just want to start tracking it immediately.

4.1.8.5 Lookup

To look up an object in the set of OCAAS databases, type its name in the text field to the right of “Source name:” and press Enter (or click on Lookup). If the name is found, all of its position values at the current moment and location will be computed and displayed in the fields just below. The databases are the files ending in .edb which reside in $TELHOME/archive/catalogs. The major planets are treated as a special case internal to OCAAS.

4.1.8.6 Track

Clicking on this button will drive the telescope to the position indicated in the RA, Dec and Epoch fields, in the lower section of this region, unless these are beyond the current limits. Once there, the telescope will then start to track this position until a new telescope command is issued or a limit is encountered. If these fields are not set but others are, they will first be computed as described in the section about the calculator ability of this region.

If a numerical value for the Epoch is shown, then the values for RA and Dec are astrometric topocentric, that is, precessed to the given epoch and corrected for parallax. If the value for Epoch is shown as EOD, then the RA and Dec are apparent topocentric, that is, precessed to the current moment, and corrected for relativistic Solar deflection, nutation, aberration and parallax.

4.1.8.7 Calculator

Values can be entered for subsets of these various quantities in the lower section, and computed for the remaining quantities by pressing Enter (or clicking on Lookup). Valid sets to enter are Alt and Az; HA and Dec; and RA, Dec and Epoch. To help enforce these sets, starting to type in any field will clear out the fields which will be computed, and leave the fields alone which belong to the same set. Related to this effect, you will find in order to change a field completely, you must first select it with the cursor then delete it then type in the new value; unlike most other text fields which automatically delete whatever is selected if you just start typing.

4.1.9 Site Information

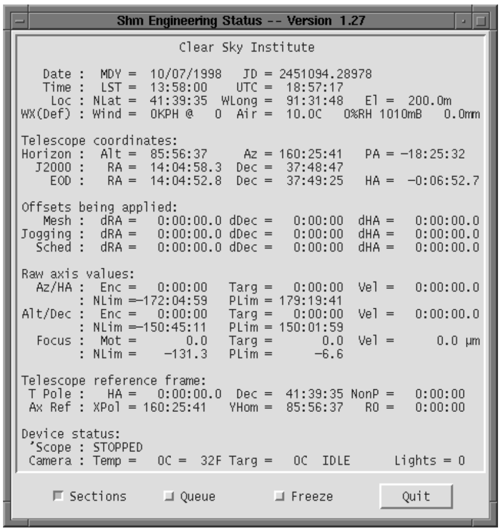
Across the middle portion of xobs are many read-only fields which display timely information about the observing site. The location of the site is defined by the LATITUDE and LONGITUDE parameters in telsched.cfg. The name of the site is the BANNER parameter. Dawn and Dusk refer to when the sun is SUNDOWN below the horizon. Weather stats are shown if a weather daemon is active; fields are shown in red if they exceed a limit defined in wx.cfg. The right three fields, T1 - T3, display the auxiliary temperature sensor values, if configured.

4.1.10 Messages

The bottom portion of xobs displays timely messages about system activity. Watch this area for information. New messages are always entered at the bottom of the list, and the list automatically scrolls to keep the bottom in view. Each entry is stamped with the UT time. The list may be scrolled, erased or moved to the bottom using the controls at the right. All entries which appear here are also stored in $TELHOME/archive/logs/xobsmsgs.log.

4.2 Shm

This program is a passive display of all major system functions. By ‘‘passive’’ is meant no hardware or other OCAAS commands can be issued from this dialog. As many simultaneous instances of *shm* man be running as desired. The display is updated at frequent intervals to show the current time and telescope, camera, dome and batch activities. The following explains each field displayed by *shm.* Some rows man not be present, depending on the hardware configuration. The rows can be broken into related sections with their own subtitle if desired by pressing on the *Sections* toggle at the bottom of the dialog.



Date — The current Month/Day/Year and Julian date.

Time — *LST* is the Local Sidereal Time, i.e., the Right Ascension currently crossing the meridian; in D:M:S. *UTC* is the current Universal Time, in D:M:S. If the GPS system is installed and operating, these values will be controlled by the GPS receiver. Otherwise, the time and date are based on the computer clock.

Loc — The geographic location of the telescope. *NLat* is the geographic latitude of the telescope, in D:M:S, with positive values indicating north. *WLong* is the geographic longitude of the telescope, in D:M:S, with positive values indicating west. If the GPS system is installed and operating, these values will be set from the GPS receiver. Otherwise, the values are simply read from the *telescoped.cfg* configuration file. *El* is the site elevation in meters.

Wx — This row indicates the current weather data. If the Weather station system is installed and operating, these values are updated automatically from it. Otherwise, only the Pressure and Temperature values are set, and these are simply read from the *telescoped.cfg* configuration file and the label is WX(Def). If any weather parameters are beyond limits, the label changes to WXALERT.

AuxTemp — Displays the current temperature readings at the auxiliary sensors. Up to three sensors are supported. These are configured in wx.cfg.

Horizon — The current Altitude and Azimuth of the telescope, in D:M:S. Altitude is positive up from the horizon; Azimuth is positive turning east from north. Also shown is the position angle, PA, defined as the angle Zenith-Telescope-Pole, positive when west of the meridian, in D:M:S. For those systems with field rotators, this angle plus the rotator offset (R0) defines the angle of the rotator with respect to its home position.

J2000 — The current Right Ascension and Declination of the telescope, precessed to 2000. Also known as the Astrometric Place of the telescope.

EOD — The current Right Ascension and Declination of the telescope, precessed to the Epoch of Date and corrected for aberration, nutation, deflection and refraction. Also known as the Apparent Place of the telescope. Also shown is the Hour Angle of the telescope, in H:M:S.

(Target)	If the telescope is currently slewing to, acquiring or tracking an object, this row contains the Right Ascension and Declination of the target object, precessed to Epoch of Date and corrected for apparent place. The left column contains up to the first seven characters of the name of the target object, or a generic anonymous name if the target was not derived from a catalog entry. An asterisk (*) appears to the right of the name if an offset is currently being applied to the object's coordinates.
Mesh	Displays the pointing corrections being applied from the *telescoped.mesh* pointing mesh file.
Jogging	Displays the offsets currently being applied to the target object location due to the Paddle. An asterisk (*) appears if the offsets shown are part of the Target.
Sched	Displays the offsets currently being applied to the target object position due to the RAOFFSET and DECOFFSET keywords in a batch schedule request. An asterisk (*) appears if the offsets shown are part of the Target.
Az/HA	Raw data regarding the longitudinal axis of the telescope, that is, the Polar axis of an equatorial mount, or the Azimuth axis of an altazimuth mount. *Enc* is the current canonical angle from the home position, in D:M:S; *Targ* is the desired position; *Vel* is the commanded motor velocity, in D:M:S per second. *NLim* is the angle from the home to the canonical negative limit; *PLim* is the angle to the positive limit.
Alt/Dec	Raw data regarding the latitudinal axis of the telescope, that is, the Declination axis of an equatorial mount, or the Altitude axis of an altazimuth mount. *Enc* is the current canonical angle from the home position, in D:M:S; *Targ* is the desired position; *Vel* is the commanded motor velocity, in D:M:S per second. *NLim* is the angle from the home to the canonical negative limit; *PLim* is the angle to the positive limit.
FldRot	Raw data regarding the field rotator of the telescope. *Mot* is the current canonical angle from the home position, in D:M:S; *Targ* is the desired position; *Vel* is the commanded motor velocity, in D:M:S per second. *NLim* is the angle from the home to the canonical negative limit; *PLim* is the angle to the positive limit.
Focus	Raw data regarding the focus motor of the telescope. All units are in microns (µm). *Mot* is the current canonical distance from the home position; *Targ* is the desired position; *Vel* is the commanded motor velocity. *NLim* is the distance from the home to the canonical negative limit; *PLim* is the distance to the positive limit.
Filter	Raw data regarding the filter wheel. *Mot* is the current canonical position from the home position, measured as an angle, in D:M:S; *Targ* is the desired position, in D:M:S; *Vel* is the commanded motor velocity, in D:M:S per second. To the immediate right of the velocity is the current filter wheel designation, or the symbols "<" or ">" if the filter is moving to a new position.
T Pole	This row and the next contain coefficients for the current telescope basic pointing model. *HA* is the Hour Angle of the telescope polar axis, in H:M:S; *Dec* is the Declination of the telescope polar axis, in D:M:S; *NonP* is the nonperpendicularity angle (the angle by which the latitudinal axis deviates from being exactly 90 degrees from the longitudinal axis).
Ax Ref	This row and the previous contain coefficients for the current telescope basic pointing model. *XPol* is the canonical angle of the longitudinal axis home to the North celestial pole, in D:M:S; *YHom* is the canonical angle from the longitudinal plane of the telescope to the latitudinal axis home switch, in D:M:S; *R0* is the

	canonical angle from the home position of the field rotator to that angle which yields images with North up when the telescope is pointing along the meridian.
Scope	Current status of the telescope mount. *STOPPED* indicates that no mount motors are currently active. *HOMING* indicates at least one motor is seeking its home switch position; *LIMITING* indicates at least one motor is seeking the locations of its two limit switches; *SLEWING* indicates the mount is making a high speed move to a new location at which it will stop; *HUNTING* indicates the mount is making a high speed move to a new location at which point it will begin tracking an object; *TRACKING* indicates the mount has locked onto and is tracking an object. Following this state information will appear *ZENFLIP* if it is enabled in telescoped.cfg. If telescoped.cfg indicates the mount is a German Equatorial, the keyword *GERMEQ-Flip* or *GERMEQ-NoFlip* will be present, depending on whether the mount is looking at the western or the eastern sky.
Dome	Current status of the Dome, Shutter and/or Roof observatory housing. If the status is preceded with *AUTO* it means the dome slit is automatically tracking the pointing direction of the telescope mount.
Camera	Current status of the CCD camera. *IDLE* indicates the camera is not currently in use; *EXPOSING* indicates the camera is acquiring an image or taking a calibration frame; *DIGITIZING* means the computer is currently downloading pixel data from the camera. Also shown are the current and target cooler temperatures. If automatic flats are enabled via the *MAXFLINT* parameter in telescoped.cfg, a number to the right will indicate the current lighting state.

4.2.1 Batch queue status

The lower portion of the *Shm* window can indicate the current status of the background batch scheduling system. This information is only displayed if the *Queue* toggle button at the bottom of the dialog is pressed in. The batch system can basically be in one of three states. One state is when there are no observations queued at all, either now or in the future. Another state is when the next observation occurs at some future time. The third state is when an observation is currently in progress. Details of these conditions are displayed, including when the next observation will begin or the current observation began, as the case may be. Also indicated is whether the background state execution process, known as *telrun*, is running. If *telrun* is not running, batch scheduled operation is not possible. *Telrun* is started and stopped using *xobs.* It can also be started automatically at boot time for fully autonomous operation.

4.2.2 Freeze frame

The information displayed by *shm* is updated very often. Occasionally it is interesting to stop the display in order to study various entries carefully. This may be accomplished in two ways. One is to press the *Freeze* toggle at the bottom of the dialog. Releasing this toggle will resume automated updating. The display may also be frozen by using the left mouse button and dragging over any portion of the display. This also copies the information selected so it may be pasted into another window (using the middle mouse button). Freezing the display in this way is handy for capturing certain values into a text file or report. Note that this causes the Freeze button to be pressed. Pressing the button again to toggle it off will resume updates.

5 Scheduled Operation

OCAAS allows the telescope to carry out a series of observations completely unattended. The series is built from a sorted collection of individual *observing requests*. Each request is defined in a separate file with the extension *.sch.* The format of this file is defined in this section.

Once the collection of individual requests is defined, they are read into an interactive sorting program called *telsched.* This program reads many individual request files and combines them in the form of a sorted set of *scans* to form one schedule for the telescope in advance. These scans are stored in a scan list file named *telrun.sls* in $TELHOME/archive/telrun. Schedules may be prepared days in advance and left standing-by.

The *telrun* program serves as the basic automated sequencer for the system when running automatically. It continually scans the telrun.sls file for new commands. To change the set, just change the file. To stop automated operation, remove (or rename) the file.

5.1 Observing Requests in *.sch* Files

Each observing request is defined in its own file, with the extension .sch. The format of the file consists of collections of *Keyword=Value* pairs. The equal sign (=) is optional. Any number of pairs may be present, although some pairs interact and so must be used correctly. Keywords retain their values until changed. Case is ignored in all keywords, and also for those values which are chosen from a specific predefined set. Values which contain whitespace must be surrounded by single quotes (‘). Comments may occur anywhere and consist of all characters from any occurrence of the pound sign (#) or bang (!) to the end of the same line. Blank lines may appear anywhere and are ignored.

Once the desired keywords have been defined the observation is marked as completely defined by a slash (/). Since the keywords support creating repeated observations in a loop, the slash often appears as the last entry in each loop.

These files may be created manually using any text editor such as *vi* or *xedit.* Or they may be created using a GUI application named *mksch* or using an Internet Web form interface. They are then combined into an overall aggregate batch sequence for an entire night using the *telsched* program.

Follows is a description of each keyword recognized in the *.sls* file format.

BINNING — The number of CCD pixels which are to be combined within the camera to form one pixel in the resulting file. The format is the number of pixels to bin in the X and Y dimensions, separated by a comma (,). For example, 2,4 means to combine 2 pixels in the X dimension to form one in the file, and 4 in the Y dimension. If this keyword is not specified, the default value is “DEFBIN,DEFBIN”, where DEFBIN is from *telsched.cfg*.

BLOCK — Introduces a group of commands. The commands themselves may result in one or more scans. Then the group as a whole may be repeated any number of times. The value associated with this keyword is the time interval between repeating each block, in H:M:S. The group is terminated with the BLOCKREPEAT keyword.

BLOCKREPEAT — Closes a group of commands started with the BLOCK keyword which are forming one set of observations. The value is the total number of times the group is repeated.

CCDCALIB — This determines whether calibration frames, data, or both will be taken. The other schedule request parameters apply to the data only (if any). The calibration frames are taken according to the values of NBIAS, NTHERM, THERMDUR and NFLAT in camera.cfg. Valid values for this parameter are:

NONE	No corrections, just take a raw data frame
CATALOG	Take data, then apply latest correction files from archive/calib.
BIAS	Take a new BIAS frame first, store in archive/calib, then proceed as with CATALOG
THERMAL	Take new BIAS and THERMAL frames first, store in archive/calib, then proceed as with CATALOG
FLAT	Take new BIAS, THERMAL and FLAT frames first, store in archive/calib, then proceed as with CATALOG
BIAS ONLY	Take just a new BIAS frame, store in archive/calib. No data is taken.
THERMAL ONLY	Take just new BIAS and THERMAL frames, store in archive/calib. No data is taken.
FLAT ONLY	Take just new BIAS, THERMAL and FLAT frames, store in archive/calib. No data is taken.

COMMENT — A comment to be added to the FITS header of the image created by this observation. If the comment value includes any blanks the value must be surrounded by single quotes.

COMPRESS — The amount of compression to be applied to the image. A value of 0 will disable compression. A value of 1 will apply a lossless compression which will usually reduce the size of the file by approximately a factor of two. Larger compression values will result in larger reductions in file size but the at the expense of not being able to exactly recreate the original pixels. The compression uses is the H-Compression algorithm developed at the Space Telescope Science Institute. Values up to about 100 allow the file to be restored with suitable accuracy to leave quantitative photometric and astrometric largely unaffected, although at progressively significant loss in aesthetic value for visual usage. If this keyword is not specified the default value is as defined by the COMPRESS parameters in the telsched.cfg config file.

DEC — Declination of the image center. The format is DD:MM:SS.S, optionally preceded by a minus sign (-) to indicate the southern hemisphere. Setting this keyword cancels any pending SOURCE value, and also requires that RA and EPOCH be set before the current observation definition is closed.

DECOFFSET — Value to be added to DEC. This keyword is intended to be used when the target is specified using the SOURCE keyword and the target moves, such as comets, asteroids or planets. The format is DD:MM:SS.S, optionally preceded by a minus sign (-).

DURATION — The length of the exposure, in seconds. The format permits any number of values to be specified, each separated by a comma. If more than one FILTER is specified, the durations are paired with the corresponding FILTER. If there are more FILTERs than DURATIONs the last DURATION is assigned to all subsequent FILTERs. If there are more DURATION values specified than FILTERs, the last FILTER is assigned to all subsequent DURATIONs.

EPOCH — The reference frame for the RA and DEC keywords, in decimal years.

FILTER — The code designating the filter. Valid codes are as listed in the *filter.cfg* configuration file. The format allows any number of codes to be specified, each separated by a comma. Generally speaking, there should be as many FILTERs specified as DURATIONS. See the DURATION keyword for what happens when the number of each does not match.

HASTART — Requests that the observation be scheduled so that the SOURCE (or RA/DEC position) is located at the given local Hour Angle when the observation begins. The format is HH:MM:SS.SS. The LSTDELTA keyword specifies the allowed tolerance. If this keyword and LSTSTART are not specified, the target will be scheduled as near transit as possible.

IMAGEDIR — The directory in which the final image file will be stored. The default is $TELHOME/user/images.

LSTDELTA — The allowed tolerance for a specified HASTART or LSTSTART. The format is HH:MM:SS.S. If the observation can not be scheduled within this tolerance, it will not be performed at all. If this value is not specified, it is defined from the parameter LSTDELTADEF in the *telsched.cfg* config file.

LSTSTART — Requests that the observation be scheduled to begin at the given Local Sidereal Time. The format is HH:MM:SS.SS. The LSTDELTA keyword specifies the allowed tolerance. If this keyword and LSTSTART are not specified, the target will scheduled as near transit as possible.

OBSERVER — Specifies the value of the OBSERVER string keyword to be added to the FITS header. Remember that the value must be surrounded by apostrophes (‘) if it contains any spaces.

PRIORITY — An integer which influences the scheduling algorithm. The scheduler sorting algorithm satisfies observing requests in increasing order of PRIORITY, that is, lower values are done first and so are more likely to be satisfied. If this keyword is not specified, the default value is 100. The valid range is -32768 .. 32767.

RA — Right Ascension of the image center. The format is HH:MM:SS.S. Setting this keyword cancels any pending SOURCE value, and also requires that Dec and EPOCH be set before the current observation definition is closed.

RAOFFSET — Value to be added to RA. This keyword is intended to be used when the target is specified using the SOURCE keyword and the target moves, such as comets, asteroids or planets. The format is HH:MM:SS.S, optionally preceded by a minus sign (-).

REPEAT — Number of times to repeat the current scan definition. Each repetition will repeat immediately. To repeat with a delay between scans, use BLOCKREPEAT. May also be used within a BLOCK/BLOCKREPEAT set to schedule a collection of scans separated by a time delay.

SHUTTER — Specifies how the camera shutter is controlled while taking data. The valid values are as follows. If not specified, the default is 1. If not data is being taken because CCDCALIB is one of the ONLY options, then this value is not used.

0	Shutter will remain closed
1	Shutter will be open throughout the DURATION
2	Shutter will be open for DURATION/2, then closed for DURATION/4, then open for DURATION/4.

SOURCE — The name of the object to be imaged. The name must occur within a catalog file in $TELHOME/archive/catalogs/*.edb, ignoring spaces and case. As a special convenience, asteroids may be specified with just their numeric designation.

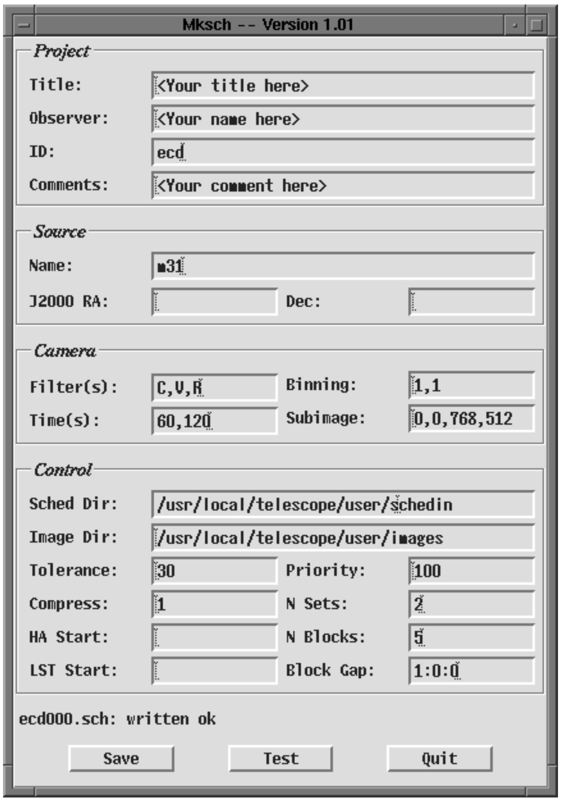
SUBIMAGE — The number of CCD pixels to use for the image. The format of this value is four integers separated by commas. The four numbers are the X and Y coordinates of the upper left corner of the image and the width and height of the image. If not specified, the default is “0,0,DEFIMW,DEFIMH”, from *telsched.cfg* .

TITLE — Specifies the value of the TITLE string keyword to be added to the FITS header.

UTDATE — The UTC date at which the observation will be scheduled. *Telsched* will skip this entry if it does not match the date for which it is building a scan list. The format is MM/DD/YYYY.

5.2 mksch – a tool for creating an observing request

This interactive program is used to create one observing request file. It presents the user with a simple form in which the various options and values may be easily entered and selected. When the form has been filled out as desired, it can be checked for self-consistency and saved to disk for later inclusion by telsched to build a plan for an entire night.



Not all capabilities available with the .sch request format are supported through this interface, particularly regarding complex sets within blocks and scheduling camera correction images. These requests will require some manual editing of the .sch file.

Follows is a description of the various fields, grouped by section as they appear on the form.

5.2.1 Project

This section captures the title, observer, user ID code and comments. (Corresponds to the TITLE, OBSERVER and COMMENT .sch keywords; the ID is used to construct a new unique .sch filename).

5.2.2 Source

This section specifies the source to be observed. If the target object is contained in one of the many OCAAS databases, then its name is all that is required. Otherwise an arbitrary J2000 RA and Dec may be entered. In this case, a Name must still be entered to identify the scans as they are taken later. (Corresponds to the SOURCE, RA, DEC and EPOCH .sch keywords).

5.2.3 Camera

This section specifies the camera settings for this schedule. The filter, duration, binning and subimage values may be specified.

The Filter and Duration fields work together to form a *Set.* Several filter codes and durations may be specified, separated by commas. They will be paired to create one exposure each. If there is more of one than the other specified, the last value for the entry with fewer values will be repeated as necessary. In the example shown, the set consists of three exposures: one for 60 seconds using the C filter and two more each for 120 seconds using the V and R filters. (Corresponds to the FILTER and DURATION .sch keywords).

The Binning field requires two values separated by a comma. They specify the horizontal and vertical pixel binning, respectively. The default binning in both directions is DEFBIN from telsched.cfg. (Corresponds to the BINNING .sch keyword).

The Subimage fields requires four values separated by commas. They specify the X and Y position of the upper left corner of the subframe with respect to the upper left corner of the CCD, and its width and height. All values are in pixels. The default X and Y are 0, and the default width and height are DEFIMW and DEFIMH from telsched.cfg. (Corresponds to the SUBIMAGE .sch keyword).

5.2.4 Control

This section collects several additional scheduling control features together.

Sched Dir specifies the directory in which the new .sch file will be saved. The default is the correct place for coordinating with other OCAAS tools.

Image Dir specifies the directory in which the images which result from this schedule request will be saved. The default is the correct place for coordinating with other OCAAS tools. (Corresponds to the IMAGEDIR .sch keyword).

Tolerance specifies the acceptable scheduling window for this request. The format is in minutes. If the system is heavily subscribed, requests with smaller tolerances at a given Priority are more likely to be dropped when sortied by *telsched.* This field is only used if HA or LST Start are specified. (Corresponds to the LSTDELTA .sch keyword).

Compress specifies the scale factor by which the final images will be compressed. Note this is not simply the factor by which the file will be reduced in size. A value of 0 requests no compression. A value of 1 requests the maximum amount of compression possible without loss of data. Values up to approximately 100 retain good photometric and astrometric quantitative data but look too blocky for use as "pretty pictures." The algorithm used is H-Compress from STScI. (Corresponds to the COMPRESS .sch keyword). Values near 10 work well for most cases.

HA Start and *LST Start* specify the start time for each set. Either can be used depending on what constraints your project requires. Only one may be specified; leave the other blank or leave both blank if no timing constraints are required. The accuracy with which the set must be scheduled to the value given is specified with the Tolerance parameter. (Corresponds to the HASTART and LSTSTART .sch keywords).

Priority is used by the algorithm in *telsched* as the first sorting criteria. The entire collection of requests are first sorted by Priority. All requests at a given Priority are treated as a competing collection. Collections with smaller numeric Priority values are assigned observing times before those with larger values. This field effectively becomes meaningless if all requests use the same value or, to put it in a more positive light, all requests will be sorted fairly if they all have the same Priority. This field is an integer, and negative values are allowed. (Corresponds to the PRIORITY .sch keyword).

N Sets specifies the number of times the exposures specified in the Camera section will be repeated back-to-back. In the example shown, the three exposures will be repeated once for a total of 2 sets or 6 exposures. These 6 exposures are called a *block.* All exposures in a block will attempt to be sorted as closely together as possible but the order implied by the Filter and Duration fields may not be preserved. (Corresponds to the REPEAT .sch keyword).

N Blocks specifies the number of times the entire block of exposures will be scheduled. In the example shown, the set of 6 exposures will be scheduled a total of 5 times separated by one hour, so this request is asking for a total of 30 exposures. (Corresponds to the BLOCK .sch keyword).

Block Gap specifies the interval between the starting times of each successive block. The format is H:M:S. In the example shown, each block will begin one hour apart. (Corresponds to the BLOCKREPEAT .sch keyword).

5.2.5 Sample output

Based on the sample above, the following .sch file is generated:

```
! Generated by mksch Version 1.01
! By Observer ID ecd
! On UTC Thu Oct  8 16:52:10 1998


TITLE = '<Your title here>'
OBSERVER = '<Your name here>'

BLOCK = '1:0:0'
    SOURCE = 'm31'
    FILTER = 'C,V,R'
    DURATION = '60,120'
    BINNING = '1,1'
    SUBIMAGE = '0,0,768,512'
    LSTDELTA = '30'
    PRIORITY = 100
    COMPRESS = 1
    IMAGEDIR = '/usr/local/telescope/user/images'
    COMMENT = '<Your comment here>'
    REPEAT = 2
    /
BLOCKREPEAT = 5
```

5.2.6 Checking and Submitting

The buttons along the bottom serve to process the request specified with the fields above. To test a request for basic requirements and consistency, press the button labeled *Test.* Any errors found will be reported in the message line immediately above the row of buttons. Since the message area is only one line, only one error is reported at a time. Fix that and resubmit. When no problems are found a message will indicate the schedule is good.

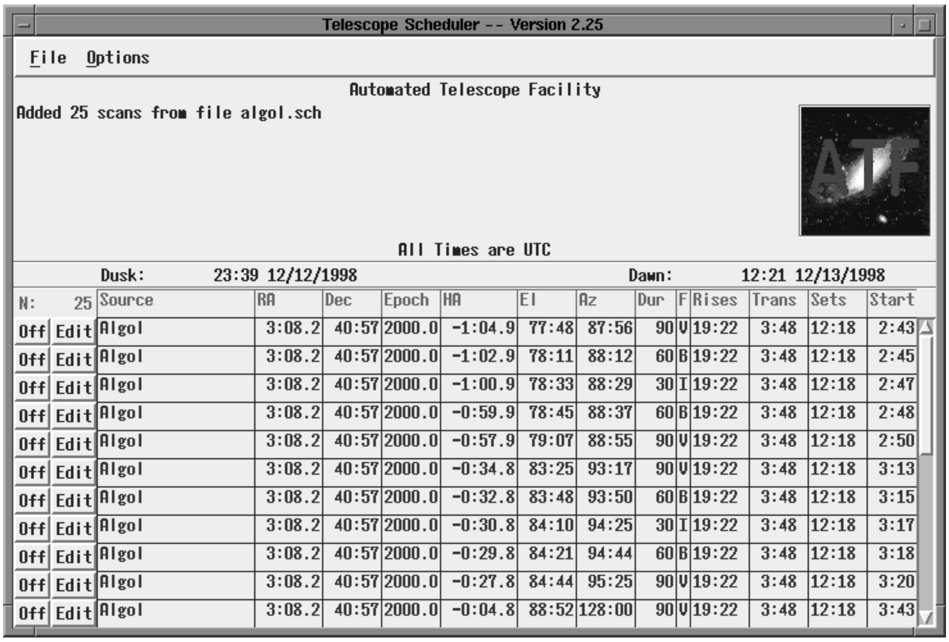
To save the request for scheduling, press the button labeled *Save.* This also subjects the request to the same tests and the submission will not be performed if any errors are detected. If the submission is successful, the file name of the .sch file will be reported.

5.3 telsched – combining many *.sch* requests into one *.sls* scan list

This interactive program is used to read many individual observing request files and form one overall sequence of events for an entire night. The program can sort the requests and tries to satisfy as many as possible while at the same time minimizing the amount of slewing required by the telescope. A graphical timeline can be displayed which shows the utilization of the night at a glance and marks conflicts. The user may add, delete and edit requests to resolve conflicts manually or let the sorting algorithm satisfy requests automatically according to priority.

Once a final overall schedule has been produced, it is saved as a sequence list file in *.sls* format for use by the *telrun* program. Schedule lists may be produced for individual days arbitrarily in advance.

Individual schedule request files, .sch, are expected to be in $TELHOME/user/schedin. The resulting can list file for telrun is expected to be $TELHOME/archive/telrun/telrun.sls.



		Source	RA	Dec	Epoch	HA	El	Az	Dur	F	Rises	Trans	Sets	Start
Off	Edit	Algol	3:08.2	40:57	2000.0	-1:04.9	77:48	87:56	90	V	19:22	3:48	12:18	2:43
Off	Edit	Algol	3:08.2	40:57	2000.0	-1:02.9	78:11	88:12	60	B	19:22	3:48	12:18	2:45
Off	Edit	Algol	3:08.2	40:57	2000.0	-1:00.9	78:33	88:29	30	I	19:22	3:48	12:18	2:47
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:59.9	78:45	88:37	60	B	19:22	3:48	12:18	2:48
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:57.9	79:07	88:55	90	V	19:22	3:48	12:18	2:50
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:34.8	83:25	93:17	90	V	19:22	3:48	12:18	3:13
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:32.8	83:48	93:50	60	B	19:22	3:48	12:18	3:15
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:30.8	84:10	94:25	30	I	19:22	3:48	12:18	3:17
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:29.8	84:21	94:44	60	B	19:22	3:48	12:18	3:18
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:27.8	84:44	95:25	90	V	19:22	3:48	12:18	3:20
Off	Edit	Algol	3:08.2	40:57	2000.0	-0:04.8	88:52	128:00	90	V	19:22	3:48	12:18	3:43

The sample screen above was the result of reading the following .sch file and sorting:

```
# Intro
title = 'Monitor Algol'
observer = 'John Doe'

# Object
epoch = 2000
ra = 3:8:10.1
dec = 40:57:20
source = Algol

# block of 5 images every 30 mins repeated 5 times = 25 images
block 0:30:0
    # b for 60s, v for 90s done twice = 4 images
    filter = b,v
    duration = 60,90
    repeat 2 /

    # 1 of i for 30s done once = 1 image
    filter = i
    duration = 30
    repeat = 1 /
blockrepeat 5
```

5.3.1 Main layout

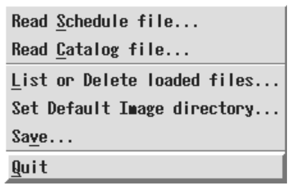
The main window of telsched is divided into two sections. The top section offers the menubar of controls, described in the next sections; a title row to display the name of the facility being scheduled; a region to display information messages; a frame on the right to display an image file; and the times of dusk today and dawn tomorrow. The facility name is set with the BANNER parameter in the telsched.cfg config file. The image file displayed must be in X pixmap format and be named as $TELHOME/archive/logo.xpm. (Hint: You can use *xv* to convert most file formats to .xpm).

The lower section is a scrolled list of the individual scans which resulted from exploding each schedule request file. Info displayed include its equatorial and horizon coordinates at the proposed time of the scan; the duration of the scan and filter to be used; the UTC times when the object rises, transits and sets; and the assigned time the scan is to begin. If the schedule request file did not specify a starting time, and the scans have not yet been sorted, then the time is not yet specified and this will be indicated by ‘‘Any’’.

A scan is drawn in green if it is eligible and is not manually turned off, otherwise it is drawn in red. If a starting time has not yet been assigned to the scan, Eligible just means the object is visible sometime today. If a time has been assigned, it means that at the proposed time for the scan the object is up, is positioned within the safe operating limits of the telescope, and the time is between dusk and dawn. The latter constraint can be disabled by setting the IGSUN parameter in the telsched.cfg config file. Telescope motion constraints are set using the parameters MINALT, MAXALT, MAXHA and MAXDEC in the telsched.cfg config file.

To the left of each scan are two buttons. The button labeled Off serves two purposes. Before the scans are sorted, this button may be clicked on or off to remove an individual scan from being considered for sorting, as desired for example when working out conflicts. If, after sorting, a scan can not be fit in for any reason, this button will be pushed in and will not respond to mouse clicks. The button labeled Edit brings up a dialog which lists all parameters describing this scan and allows them to be modified. If the scan was turned off by the sorter, there is also a line to indicate why it was turned off.

5.3.2 File menu



Read Schedule file opens a file selection dialog. The pattern is set to match files which end with .sch in the $TELHOME/user/schedin directory, the default naming convention and directory for schedule requests. After a schedule file is selected, you can choose *Append* or *Edit.* Append will read the file and explode it into the scans it describes, adding them into scans already in memory. If any scans in the .sch file are marked with a UTDATE keyword that does not match the date displayed by telsched, they will silently not be included in the set of scans created in telsched. If there are any errors in the schedule request, they will be described in the main message line of telsched and no part of the file will be loaded. This is where Edit comes it. It opens a text editor with the selected schedule request file already loaded. After making changes to the file, it can be saved and attempted to be reloaded. The editor which is launched is *xedit* by default or whatever is set in the EDITOR environment variable.

Read Catalog File allows any OCAAS catalog to be loaded and one scan will be created for each object. The default place for these catalogs is $TELHOME/archive/catalogs and any file ending with the .edb OCAAS database format may be used. This feature is handy, for example, to build a scan list for all Messier objects or comets. The default filter is FDEFLT in *filter.cfg.* The default binning is DEFBIN; image size is DEFIMW and DEFIMH; starting time tolerance is LSTDELTADEF; default compression scale is COMPRESS; all in *telsched.cfg.*

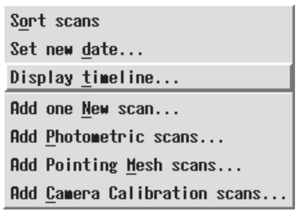
List or Delete Loaded Files brings up a dialog which lists each schedule request file which has been loaded into memory. Sometimes just seeing the list is helpful enough, since one .sch file can explode into many individual scans. A file may be selected and deleted, which removes all scans it created from the current set. There is also a convenience button to remove *all* files (and hence all scans) from memory.

Set Default Image Directory allows you to specify the directory where images which result from the scans being designed will be stored. By default, this is $TELHOME/user/images.

Save is used when you are satisfied with the new sequence list and want to save it for use by telrun as a batch program. This dialog displays the file name of the telrun.sls file which will be created; it can be changed if desired. The default is $TELHOME/archive/telrun/telrun.sls. Note that if it is renamed, telrun will not find it automatically (unless it too is run manually to change where it looks for a scan list file). The format of the .sls file is text but is not described here because it is not intended to be edited by humans. But telsched also saves the new set of scans in a form which is more suitable for human consumption known as a *summary* file. It lists each scan one per line with fixed columns. The columns are basically the same as those shown in telsched, plus a few more to indicate the source of the scan. This file uses fixed columns widths so is also suitable for import into other programs.

Quit allows you to exit the telsched program, after answering a confirmation question.

5.3.3 Options menu



Sort scans is the work horse feature of telsched. It takes the complete set of scans currently in memory that are not turned Off, and tries to build an observing sequence which performs as many of them as possible. See the separate section on the Sorting function for a description of how it works.

Set New Date brings up a dialog which allows you to enter an arbitrary date for which to design a scan list. By default, telsched chooses tonight's date unless dusk has already occurred then it uses tomorrows date.

Display Timeline brings up a dialog with a few statistics about the total sets of scans and a vertical 24-hour timeline. It is labeled in UTC down the left and in LST down the right. The labels are highlighted if they occur between dusk and dawn. Each scan currently in memory is drawn as a blue line for the interval it effects. If scans overlap, the conflict is shown to the right in red. Thus the timeline quickly shows the nights utilization and periods of over subscription. Ideas to resolve oversubscriptions include removing the entire observing request file, turning off selected scans, editing the schedule request to increase its schedule tolerance, or just letting them occur. The schedule request can include a tolerance window in which it will accept a scan outside its request range using the LSTDELTA parameter. The default value of this is set with the LSTDELTADEF parameter in the telsched.cfg config file. Telrun will try to watch for and sneak in oversubscribed scans within this range as it works through the night in an effort to fill any gaps which might arise.

Add one New scan brings up a dialog with which you may create one new scan from scratch, without the need to have it based on a .sch file. Enter all the fields as desired and Apply. You may add as many additional scans as desired from this dialog.

Add Photometric Scans is a shortcut method to insert a set of scans which will image standard Landolt photometric stars through a complete range of airmass. Each image thus created will begin with the letters *ldt.* You may specify how many fields are taken, when they will begin, and the duration for each filter. The default durations are specified in the PHOT{BVRI}DUR keywords in telsched.cfg. Each field will consist of four images, one for each of B, V, R and I filters. The fields will be spaced as evenly as possible from the zenith down to the minimum altitude. The minimum altitude is set by the MINALT parameter in the telsched.cfg config file; the maximum altitude by MAXALT. The image size and binning are set from the DEFIMW, DEFIMH and DEFBIN parameters in telsched.cfg. You may also specify where the resulting images will be stored, and the sorting priority for the scans. Telsched uses a list describing all Landolt stars stored in the file photcal.ref in $TELHOME/archive/photcal. Leaving the images in this directory will make it convenient to analyze them later using the *photcal* tool.

Add Pointing Mesh Scans is a shortcut method to insert a set of scans which will span the entire safe operating motion of the telescope down to an altitude of MINALT as specified in telsched.cfg. Each image thus created will begin with the letters *pt.* These images are expected to be used to create a mesh of pointing errors later by using the *pterrors* tool. You may specify when the mesh will begin, the spacing of the mesh grid size, the duration of each image, and the sorting priority for the scans. The usual place for these files is $TELHOME/archive/pointmesh.

Add Camera Calibration Scans brings up a dialog with which you may schedule taking Bias, Thermal or Flat camera calibration scans at any time during the night. The dialog allows specifying the directory where these image should be created. Note that if they are not placed in the standard directory (namely, $TELHOME/archive/calib) they will not be found and hence not used by subsequent scans which allow for calibration. Specifying a non-standard directory might be useful if experimental camera hardware studies are being performed for example, but usually you will want to leave the directory set to the default so the time spent making the new calibration images can benefit subsequent image correction. The dialog allows you to specify the scheduling priority, start time, and binning. Note that binning must match the binning value of each image for the calibration file to be used to correct that image. The dialog allows selecting which type or types of calibration images to create. Specifying Bias will take only a Bias. Specifying Thermal will take a Bias then a Thermal. Specifying Flat will take a Bias, then a Thermal then a Flat. The details of these calibration scans are determined by the NBIAS, NTHERM, THERMDUR and NFLAT keywords in the camera.cfg.

5.3.4 Sorting function

The sorting function breaks the time between dusk and dawn into slots of one minute each. The basic idea is to mark each slot as being available, then work through the scan list in an orderly fashion and assign each scan to one slot and mark that slot unavailable. Follows is a description of the steps involved. ‘‘Assigning to a slot’’ means assigning to a set of contiguous slots long enough to accommodate the entire duration. A scan duration depends on its DURATION keyword from the originating request file, rounded up and finessed a bit for slew time. ‘‘Rise’’ and ‘‘set’’ refers to a horizon at MINALT, a parameter in the telsched.cfg file, not the ideal horizon.

All scans are first sorted by increasing PRIORITY.

Within a priority, assign scans using the following steps in order:

1 Find scans which have a specified starting time (via the HASTART or LSTSTART keywords). Try to assign each to their exact slot. If the slot is already assigned, search in either direction for the closest available slot up to a maximum of LSTDELTA away. If none found, turn object Off and (arbitrarily) set start time to the original time requested.

2 Find remaining unassigned scans which are ‘‘evening’’ objects, that is, set after but transit before dusk. Sort by increasing time of setting. Assign each by starting at dusk and searching forward to the first available slot, up to the time of set. If none found, turn object Off and (arbitrarily) set the start time to dusk.

3 Find remaining unassigned scans which are ‘‘morning’’ objects, that is, rise before but transit after dawn. Sort by decreasing time of rising. Assign each by starting at dawn and searching backwards to the first available slot, back to the time of rise. If none found, turn object Off and (arbitrarily) set the start time to dawn.

4 Find remaining unassigned scans which are ‘‘night’’ objects, that is, transit between dusk and dawn. Try to assign each to the slot at their time of transit. If the slot is already assigned, search in either direction up to dusk and dawn for the closest available slot. If none found, turn object Off and (arbitrarily) set the start time to the time of transit.

5 Find remaining unassigned scans which are circumpolar objects. If the object transits between dusk and dawn, proceed as for ‘‘night’’ objects. If the object transits nearer dawn than dusk, proceed as with ‘‘morning’’ objects. Else proceed as with ‘‘evening’’ objects. If no suitable slot is found, turn object Off and (arbitrarily) set the start time to the time of transit.

6 All remaining scans are daytime objects. Turn them Off and (arbitrarily) set the start time to the time of transit.

5.4 telrun -- executing the plan

The *telrun* process is a daemon process which reads a schedule file in *.sls* format and carries out the directions in detail. This process is not controlled directly by the operator. This process is controlled via the *Batch* control from within the *xobs* program.

Telrun takes all of its directions from one file, $TELHOME/archive/telrun/telrun.sls. This is the file which was prepared using the *telsched* program. This file is updated in-place in real-time as *telrun* works through it. In particular, the Status line is marked N if the associated scan is New and has not yet been attempted; D if the scan is completed; and F if the scan failed for any reason. *Telrun* allows this file to be changed out from under it. If the file disappears, *telrun* will simply stop performing any actions. If the file suddenly reappears, *telrun* will begin to work through it from the top. When *telrun* begins, it always looks for this file and begins to consider the scans it contains immediately. Since the scans are time and date stamped, *telrun* knows when it may process each scan. *Telrun* begins to set up all equipment needed for the scan SETUP_TO seconds before the started starting time (SETUP_TO is in *telsched.cfg*). It then waits for the stated time to begin each scan. *Telrun* starts setting up for the next scan while downloading pixels from the previous scan. If no scans are appropriate, *telrun* does not command the telescope to perform any actions. *Telrun* does not look ahead in the file, so entries should be sorted in time order. Regardless of whether a scan is in progress, *telrun* will close the dome and stow the telescope to STOWLAT and STOWAZ when dawn occurs, unless IGSUN is set to 1.

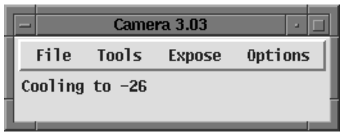
As *telrun* operates, it generates detailed log files of all of its activities. One file contains all operations and is located in $TELHOME/archive/logs/telrun.log. Telrun also generates a separate log for each observing code encountered within the .sls file it is working on. These contain only those operations which were a result of actions requested by a given observing code. These files are located in $TELHOME/user/logs.

5.5 Current queue status

There are several tools which can display information about the queue. The GUI program *shm* can display the details of the current scan, if one is in progress, or the next if one is queued to start. So can *xobs* if Batch mode is enabled. The command line programs *telshow* and *telschowl* can also do this. Telshow is particularly handy for use within cgi-bin web scripts. These programs are described in more detail elsewhere.

6 Camera -- Image Acquisition, Display and Measurement

The *camera* program serves two rather distinct roles in OCAAS. One role is direct real-time control of the CCD camera. In this role, it can command the camera to take an image and display it immediately and it can take Bias, Thermal and Flat image correction frames for use immediately or later during batch scheduled operation. The other role of *camera* is to analyze existing FITS image files using its many image analysis and measurement features.



Note that when the *camera* program is using the CCD camera the batch scheduled operation should not be active (batch operation is turned on and off using the *XObs* program). However, if one closely monitors the activity of the batch system and knows that it is not in fact using the CCD camera at the moment, it is OK to use the CCD camera from *camera* during this idle period (the current activity and time of the next scheduled scan can be monitored using the *shm* system status display and via the Batch display of *xobs*). If one is not sure, attempting to use *camera* while the CCD camera is in use by the batch system will simply result in the message that the camera is currently in use result in no harm done. However, if *camera* is using the CCD camera when the batch system wants it, the batch control process will get the in-use error and the scan will be marked as Failed and skipped.

Using *camera* purely for viewing and analyzing existing images is fine at any time, whether or not scheduled batch operation is active. It is also fine to use several instances of *camera* simultaneously in this way to analyze existing images.

As *camera* operates, status messages will appear from time to time in the top message line, just below the menubar. Watch this line for good information.

If *camera* is run on a system without a camera installed at all, the Expose menu will not be displayed.

6.1 Command-line options

Camera supports the following command line options:

-install: This causes *camera* to use a private colormap. This has the advantage that it can coexist with other programs which also use many colors, but the disadvantage that the colors are only correct when the program has the focus (such as when the cursor is within its boundaries). Basically, use this only if *camera* complains about running out of colors.

-prfb: Print each of the internal X resource defaults and exit.

-help: Print this information and exit.

Any additional arguments are assumed to be fits files to display. The first one is displayed immediately, and the remaining are added to the history list.

6.2 camera.cfg

The camera program checks for the file *camera.cfg* in $TELHOME/archive/config. This file is optional. A description of the contents of this file may be found in the Introduction section.

6.3 Expose menu: Image Acquisition

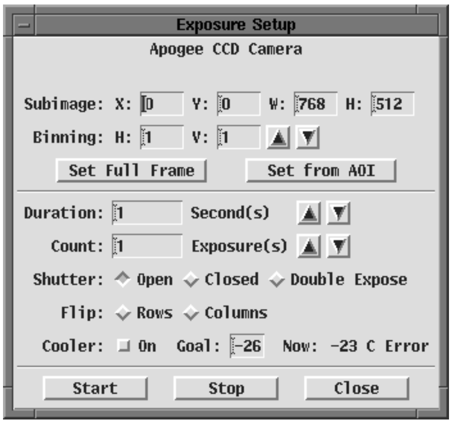
Taking an image using the CCD camera with *camera* involves first defining the exposure parameters, then taking the image. Images may be taken one at a time or a set number may be taken continuously unattended.

6.3.1 CCD driver

The path to the device driver used by *camera* is the value of the X resource *Camera.Driver,* which is dev/ccdcamera by default, located within the $TELHOME directory. This in turn is typically a symbolic link to the real device driver special file. The driver is opened and closed by *camera* as necessary; in particular, it is open whenever the Expose->Setup dialog is up and closed when this dialog is down, except during operations controlled by the shortcuts within the Expose pulldown menu itself and when the program is initially started which causes it to start the cooler.

6.3.2 Exposure setup

Set the exposure parameters by selecting the *Setup...* option from the *Expose* menu. This will bring up a dialog in which you may set image size, location, binning, duration, repeat count, shutter options, flip rows or columns, and control the cooler. The dialog will only permit settings which are valid for the installed camera. The initial settings will use the full chip at 1:1 binning. The subimage may be specified by entering desired values into the text fields directly, or by using the current size and position of the yellow AOI overlay. Binning may be entered using the test fields, or increased or decreased using the convenience arrow buttons.



In order for *camera* to access the CCD camera, it's Linux driver must already be installed. If this has been set up successfully, the title at the top of the *Exposure Setup* dialog will display the name of the camera and the initial settings will match the CCD chip size. If the camera is not available, the title will be *Unknown camera* and can not be used.

The dialog also displays the current state of the thermoelectric cooler and allows it to be turned on or off and to set its target temperature. The initial target temperature is either the value of DEFTEMP specified in the *camera.cfg* file or the current camera temperature, which ever is lower.

Note that it is far preferable to operate a CCD camera at a temperature it can hold at target rather than just setting it to drive forever towards a temperature lower than it can possibly reach. This is because at least as important as a low temperature is a constant temperature.

The toggle buttons labeled *Shutter Open, Closed* and *Double Exposure* refer to the state of the shutter at the time of the next exposure. They do not cause the shutter state to change immediately but effect the shutter operation for the next exposure. *Double Expose* means the shutter will be open for the first half of the exposure duration, closed for a quarter of the time, then open for the final quarter.

6.3.3 Taking an image

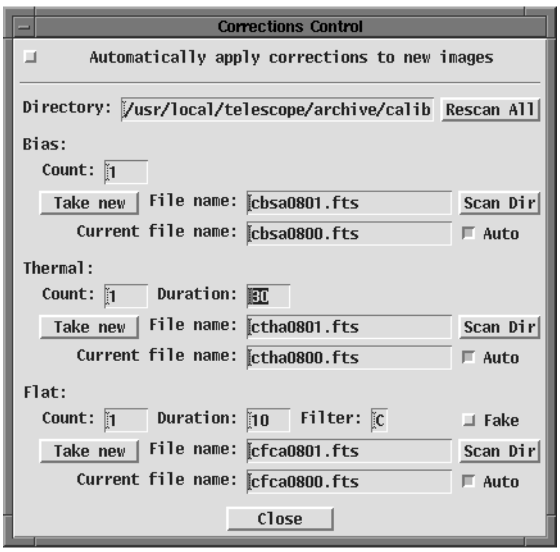
Once the exposure parameters are set as desired, set the *count* of exposures to take manually or via the convenient up and down buttons. Then pressing *Start* will command the CCD camera to begin acquiring the exposure(s). The shutter is opened for the specified time (if enabled) and a countdown is begun. The state of the camera during this period is EXPOSING. (The camera state is also displayed in the status line of *XObs* and in the Camera line of *shm.)* The seconds remaining is displayed in the message line at the top of *camera.* When the time has elapsed the state will change to READING and the pixels will be read from the camera into the host computer. If the exposure duration was for 30 seconds or longer, the computer will beep when the pixels are completed being read. While reading the pixels the system may appear sluggish. This is because the processor is nearly dedicated to reading the camera controller to insure the pixels are read as rapidly and regularly as possible for maximum image quality. (The affect can be controlled using the *impact* parameter of the driver when it is installed). When the pixel reading is compete, the state will change to IDLE and the system will again be responsive. If the *count* was greater than one, the count is decremented and another image is started immediately. The camera may be stopped at any time by pressing the *Stop* button, except during pixel reading.

Exposures may also be performed directly from the *Expose* menu off the main menu bar. Options are provided to take one image, take images continuously, or stop the camera.

If the option has been set in the *Corrections* dialog to *Automatically apply corrections*, then the most recent Bias, Thermal and Flat which match the image binning are applied. Once the image is complete, *camera* immediately displays the image. The contrast, magnification and other initial settings depend on options discussed shortly.

If the *Auto Save* option is on in the *Save* dialog, the image will also be saved to disk. Note also that, at least for all brands of CCD cameras currently supported by OCAAS, the time to read the pixels is proportional only to on the net number of rows after binning, regardless of the number of columns.

6.3.4 Taking Correction Images



Raw images taken with a CCD camera can be greatly improved if they are corrected for several artifacts. These include readout noise within the camera electronics; anomalies inherent in the chip technology including unusual pixels and the inevitable accumulation of thermal noise; and irregularities in the imaging optics such as dust on the optics and vignetting. These artifacts are measured and stored in *correction* images taken under specific conditions. They may then be applied later to raw images to largely remove the artifacts and hence "clean up" the raw images to their full potential.

The *camera* program is capable of taking, organizing and applying these correction images. There are three types of images, each designed to measure a different set of artifacts. *Camera* can save these correction images for use by the batch scheduled system, and it may also automatically apply them immediately each time an image is taken directly.

The following sections describe how each type of correction image is taken and saved. All correction images are taken using *camera* by using the dialog which is brought up by choosing the *Corrections...* option within the *Expose* menu from the main *camera* menu bar. This dialog allows setting up various options for each type of correction image (described in detail in the next sections) and allows specifying the file names which will be created and used for correction functions.

6.3.4.1 File names

Standard names are assigned to bias, thermal and flat files. Using these names will insure compatibility among all OCAAS tools. Similarly, specify the standard directory $TELHOME/archive/config to contain all correction files will insure all tools will search the same place for these important files.

All correction file names are a total of 12 characters long. These characters can be described according to the following pattern: CCCMDDNN.fts. CCC encodes the correction category as follows. Bias file names begin with *cbs;* thermals begin with *cth;* and flats begin *cfX,* where X is the filter designation (as defined in $TELHOME/archive/config/filter.cfg). M encodes the month which the file was taken, encoded a lower-case hex digit, with 1 denoting January through c denoting December. DD is the decimal day of the month, beginning with 01 and preceded with 0 if less than 10. NN is a sequence number added in case more than one file in a given category was taken on the same day. Finally, all correction file names always end with the suffix *.fts,* used throughout OCAAS to denote a FITS file. Additional information regarding the frame size, binning, duration, and other details are contained within the FITS headers of these files and is used automatically as necessary by *camera* and other OCAAS tools.

6.3.4.2 Correction Dialog Controls

The toggle button at the top of the dialog controls whether new images will be corrected. This has no effect while taking correction images.

The *Directory* text field determines where new correction files will be stored, and where in which they will be searched when needed.

Clicking any *Scan Dir* button first looks in the calibration directory for any existing files suitable for use in the corresponding role with the current image. Then it creates a new, unique name to be assigned should a new calibration file be taken, taking care to account for any existing files it finds. The name will be chosen so as to become the new default by all other places within OCAAS which choose a calibration file.

Each *Auto* toggle button selects whether the calibration directory will be scanned for the most recent image suitable for use in the corresponding role each time a new image is taken by *camera*, if on, or whether the name given to the left will be used regardless of what is in the calibration directory, if off. When first turned on, said scanning occurs once immediately and the result is displayed.

Clicking on *Rescan All* is like clicking on each *Scan Dir* button and turning each *Auto* button off and back on.

It is highly recommended that the automatically generated file names be used in all cases to insure the files are properly found by other applications within the OCAAS suite. To make sure the correct files are always being used, simply press the button labeled *Rescan All* before taking each correction image and leave the three *Auto* toggle buttons on (depressed) at all times.

A separate set of correction images must be created for each combination of row and column binning. This is because changing binning makes a significant difference in the behavior of the camera. Correction images should always be taken at full frame with no subimaging. In this way, OCAAS can extract the relevant portion when calibrating images taken with any values of subimaging.

6.3.4.3 Bias

A Bias correction image is designed to measure the signal which appears from the CCD camera even when no light has hit the CCD. It is a result of the readout noise associated with the electronics within the camera circuitry and within the CCD chip itself. To measure this effect, the idea is take an image of zero seconds duration with the camera shutter closed. In this way, the entire signal is that of the electronics alone. For statistical reasons, it is best to take several such images and average them together to get a mean value for the readout noise. The bias level is likely to change somewhat, often on time scales of hours to days. For this reason, it is wise to take new bias frames often.

Bias correction files may be taken manually via *camera*, or with the batch system.

To take a bias frame, specify the number of images to take and average together in the text field provided. Numbers between 5 or 10 are reasonable. Enter a file name for the resulting averaged bias frame or choose Auto to let OCAAS create a file name. The latter is recommended for most situations to insure the name adheres to the proper conventions and is in the proper directory for use by the batch scheduling subsystem. Pressing *Take* will then take the given number of images, average them and store a new bias correction image.

For identification purposes, each bias frame will have one FITS header string field added named BIASFR. The value of this field is ‘‘Bias frame: Averaged=xxx’, where xxx is the number of images which were taken and averaged together to create the bias image.

6.3.4.4 Thermal

A CCD chip will accumulate values from heat as well as light. Even a CCD cooled to a very low temperature is not immune. OCAAS assumes that this accumulation is proportional to the exposure time of the image.

Some systems refer to a ‘‘Dark’’ correction image. These images are created by taking an exposure of finite period but with the shutter closed. OCAAS does take Thermal images this way too, but then subtracts the Bias frame. OCAAS also stores in the FITS header of the Thermal image the exposure time of the image. When OCAAS then uses the Thermal image in later corrections, it scales the values according to the ratio of the exposure time of the correction image to that of the new image being calibrated. By removing the bias from the Thermal, this scaling is more accurate because it does not also scale the inherent bias as well.

Thermal correction files may be taken manually via *camera*, or with the batch system.

To take a thermal frame, specify the number of images to take and average together in the text field provided. Numbers between 5 or 10 are reasonable. Also specify an exposure time. Experiment with your camera to find a time which results in pixel values roughly midway from saturation, typically in the neighborhood of 30,000. (This is not critical; mean values from, say, 10,000 to 40,000 are fine). Enter a file name for the resulting averaged thermal frame or choose Auto to let OCAAS create a file name. Also specify the name of a bias frame to use or choose Auto to let OCAAS choose a suitable existing bias file. Using Auto for each step is recommended for most situations to insure the name adheres to the proper conventions and is in the proper directory for use by the batch scheduling subsystem. Pressing *Take* will then take the given number of images, average them, subtract bias, and store a new thermal correction image.

For identification purposes, each thermal image will have one FITS header string field added named THERMFR. The value of this field is ‘‘Thermal frame: Averaged=xxx Bias=yyy’, where xxx is the number of images which were taken and averaged together to create the thermal image, and yyy is the name of the bias image which was subtracted.

6.3.4.5 Flat

A Flat is intended to capture the irregularities in the optics of a camera system including scratches, dust and bubbles in the glass and vignetting along the optical path. Flats are actually just normal exposures but taken while the camera is aimed at a field which is very evenly illuminated. This allows any deviations from a perfectly ‘‘flat’ image to be attributed to the optical path irregularities and not the scene. OCAAS subtracts bias and thermal effects from images to be used as flat fields.

How to obtain an evenly illuminated reference is a matter of convenience and preference. Some people use the sky at dusk or dawn. Others build a light source which carefully diffuses the light onto a panel. Others put a white panel on the wall of their observatory and depend on ambient light to illuminate it evenly. (Dome flats may be scheduled as part of the batch operation.)

To take a flat frame, specify the number of images to take and average together in the text field provided. Numbers between 1 and 5 are reasonable. Also specify an exposure time. Experiment with your camera to find a time which results in pixel values roughly midway from saturation, that is, about 30,000. (This is not critical; mean values from, say, 10,000 to 40,000 are fine). Enter a file name for the resulting averaged flat frame or choose Auto to let OCAAS create a file name. Also specify the name of a thermal frame to use or choose Auto to let OCAAS choose a suitable existing thermal file. Using Auto for each step is recommended for most situations to insure the name adheres to the proper conventions and is in the proper directory for use by the batch scheduling subsystem. Pressing Take will then take the given number of images, average them, subtract the bias and thermal, and store a new flat correction image.

For identification purposes, each thermal image will have three FITS header field added. One is a string field named FLATFR. The value of this field is ‘‘Flat frame: Averaged=xxx Bias=yyy Thermal=zzz’, where xxx is the number of images which were taken and averaged together to create the image, yyy is the name of the bias image that was subtracted, and zzz is the name of the thermal image that was subtracted. A second field, of type real, is named FLATMEAN. It is set to the average pixel value in the entire image and improves performance later when used during flat field computations. A third field, of type string, is named FILTER and contains the single-letter designation of the filter which was in place when the flat frame was taken.

It is also possible to take a “fake” flat. This is a file with all pixels set to 1, and all necessary header fields set to satisfy system requirements for a flat correction image. This can be handy under initial setup situations.

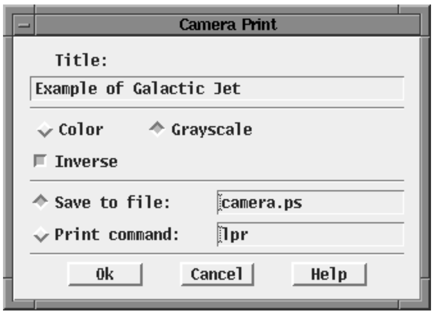
6.4 File menu

6.4.1 Opening a FITS file

This brings up a file selection dialog box from which a FITS file may be selected and displayed. Both uncompressed files, with an extension of .fts, and compressed files, with an extension of .fth, are supported.

Also, a list of the most recent several files successfully opened or saved appears at the bottom of the pull down menu, and one of these may be reopened just by selecting the desired line.

6.4.2 Printing an image



This option brings up a dialog which controls printing. The current image may be saved to a file in Postscript format, or sent directly to a printer.

Printing is actually performed by saving the image to a temporary postscript file and issuing a command to print the file. This command is typically *lpr* but a text field is provided to enter another command if desired. The name of the temporary file is appended to the text in the command line before being issued. The temporary file is deleted as soon as the command exits.

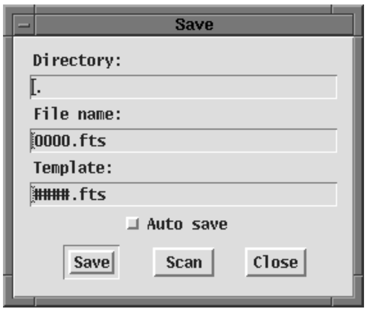
The image is formatted for a sheet of 8.5x11 inch paper, with the image on top. Below the image are several fields from the FITS header. Other options include inverting the image before printing, and to print in color (which is only useful if a pseudocolor map has been applied).

6.4.3 Deleting a FITS file

This option permanently deletes the file currently being displayed from disk.

6.4.4 Saving a FITS file

This option allows saving the current image being displayed to disk in FITS file format. This option exists in order to save an image which has just been taken directly with the CCD camera, a file to which additional COMMENT fields have been added, or to save an image as a different name.



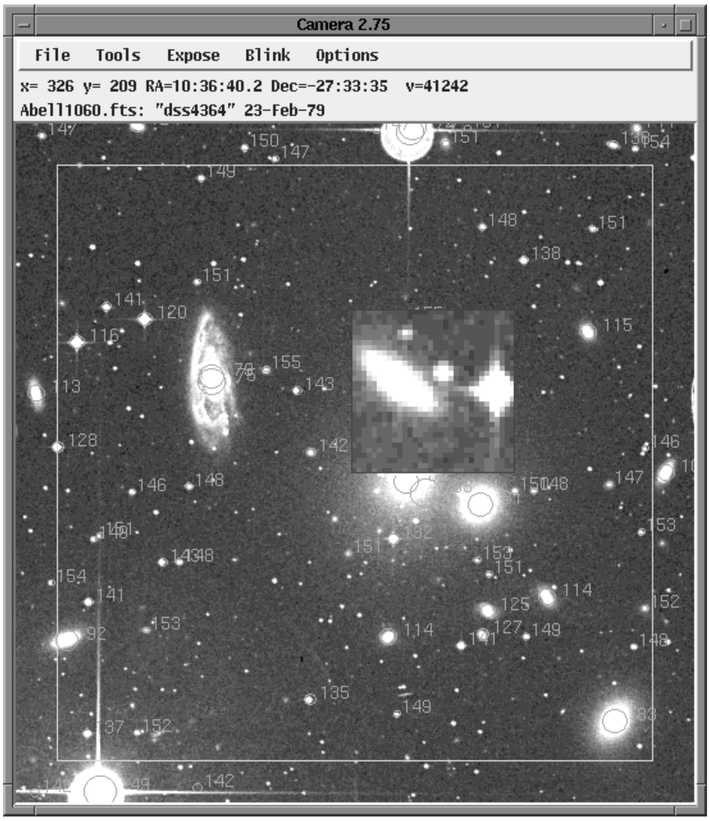
A dialog box is presented in which the directory and file name may be entered. The image will be saved with the name given whenever the *Save* button is pressed. New images acquired from the camera can be saved automatically by turning on the

Auto Save toggle button. When this is active, each new image acquired will be saved to disk, even if the Save dialog is closed.

The file name may be entered manually or created automatically based on a file name template pattern. The right-most contiguous collection of pound signs (#) in this template will be replaced by the smallest integer with that many digits which represents a non-existing file in the given directory. The template is used to create a new file name automatically each time a new image is saved whenever the *Auto Save* option is selected. This is used for saving multiple automatic exposures. The template may also be used to create a unique file name at any time manually by pressing the *Scan* button.

It is not possible to save compressed images directly. To do that, save them as usual here, then run the *fcompress* command line utility program.

6.5 Basic image display



Certain basic image control features are always active in *camera*. See the subsequent sections for features which may be turned off or on and which have their own controls.

6.5.1 Panning

The overall window size used by *camera* may be adjusted in the usual way by manipulating the resize handles surrounding the main application window. Images which are larger than the current window may be panned by using the scroll bars which will appear to the right and bottom edges of the image.

6.5.2 Image header line

Just above the image is a status line showing basic information from the FITS header of the image being displayed. This line displays, in order: the file name; the value of the OBJECT or TARGET header field; the date and time from the DATE-OBS and TIME-OBS fields; the value of the EXPTIME exposure time header field, expressed in seconds; the value of the FILTER field; the value of the ELEVATION header field, expressed in terms of airmass (labeled Z); and the values of the FWHM header fields, if present. If there are WCS header fields present to establish a pixel scale, the FWHM values are shown in arcseconds and followed by a double quote; otherwise they are shown in pixels without a quote.

6.5.3 Mouse

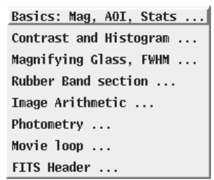
Moving the cursor over the image with the left button depressed moves a magnifying glass over the image. The magnification factor and size of the glass are controlled using a dialog available from the *Tools* menu.

Also as the cursor moves, the top line of the status area will display the pixel row and column of the cursor and the raw numeric value of the pixel. If the image header includes the traditional World Coordinate System fields (CTYPE1/2, CDELT1/2, CROTA1/2, CRPIX1/2, and CRVAL1/2) the Right Ascension and Declination of the center of the pixel are also displayed in this line. A toggle button in the Option menu called *Roaming cursor report* may be turned on to cause this coordinate display to be active even when no mouse buttons are depressed.

6.5.4 Area of interest overlay

A yellow rectangular Area of Interest, or ‘‘AOI’’, is always present as an overlay to the image. The initial default position and size of this AOI is 32 pixels in from each edge. The AOI may be redefined by using the mouse as follows. Begin by moving to the desired upper left corner of the box and pressing the middle button. Keep the button pressed and move to the desired location of the lower right corner of the AOI and release. As described in the *Tools* section, the AOI can be used to define a subset of the image for computing various statistics, cropping, and determining a brightness and contrast level. Whether this is true at any point in time is defined by the settings of these tools.

6.6 Tools menu: Image Manipulation and Analysis



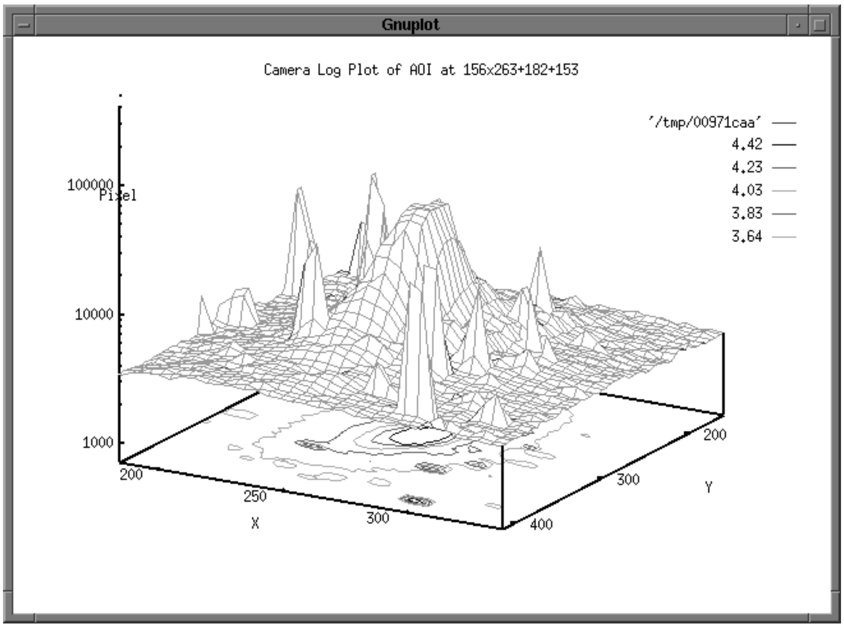
This section describes the features available via the *Tools* menu from the main *camera* menubar. Each option brings up its own dialog box from which the tool may be controlled. For the most part, all settings made while the dialogs are up remain active when the dialog is closed; exceptions will be mentioned as necessary. On the other hand, if a tool uses the mouse, the mouse will only be connected with the tool as long as the dialog is up. If the dialog for a menu selection is already up, it will be brought to the front of the stacking order.

6.6.1 Basic Operations: Size, AOI, Stats

This option brings up a dialog box offering several basic image display functions. All selections remain in effect if the dialog is closed.

The top section controls the magnification factor. These choices may be selected before the first image is opened. This is handy for setting a small magnification size while using *camera* over a slow modem connection. Next down are options for flipping the image Left-to-Right or Top-to-Bottom after it has been taken.

The lower section refers to the yellow Area-of-Interest box always overlaid on the image display. *Reset* causes the AOI to return to its original size and position, which is 32 pixels inside the image border. *3d Gnuplot* will send the pixels within the current AOI to gnuplot* as a 3d surface display.



ASCII Export will save the pixels within the current AOI to a text file. Each line of the file will contain the X, Y and raw pixel value, separated by a tab. *Crop* causes the image display to be limited to those pixels inside the AOI. Note that when cropping is turned off or Reset is activated, the magnification is automatically returned to 1x.

The next set of numbers gives the location of the upper left corner of the AOI and its width and height, in pixels with respect to the upper left corner of the image.

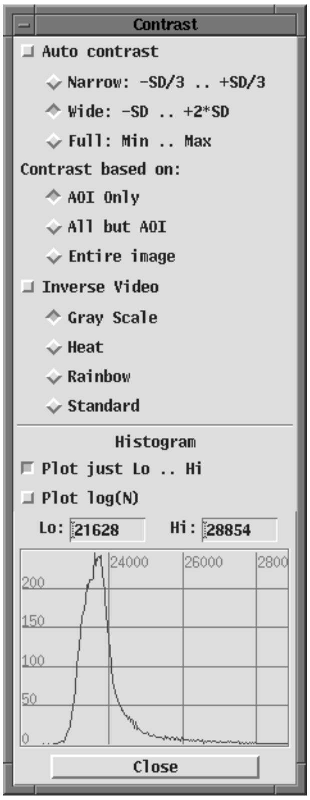
The bottom set of numbers displays basic statistics for the pixels within the AOI. These include mean, median, standard deviation, minimum, maximum, the X and Y location of the maximum pixel, and the number of standard deviations it is above the mean using a local noise annulus.

6.6.2 Contrast and Histogram

This option brings up a dialog which controls how the image pixel values mapped for display.

On top are options to control contrast. If the *Auto contrast* option is active then the contrast and brightness of the image are automatically set according to one the methods which may be selected. One method is to assign black to all pixel values which are less than or equal to the median pixel value minus one third standard deviation and white to all pixel values which are equal to or greater than the median plus one third standard deviations. This generally results in a high contrast image. A wider criteria can be selected with the next option. The third method is to assign black to the smallest pixel value and white to the brightest pixel. Of course, all options assigns pixel values in between to successive linear values of gray.

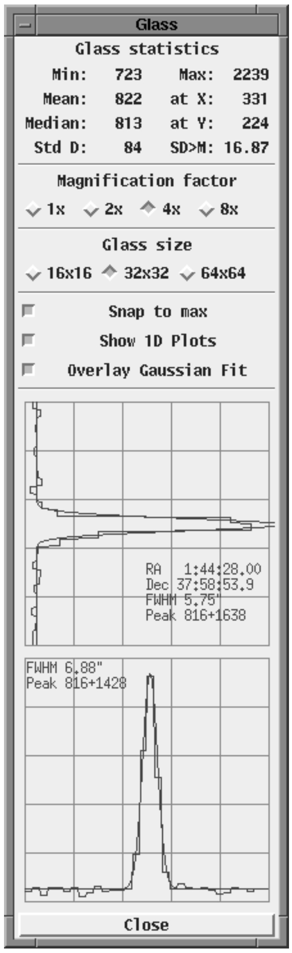
*gnuplot is a standard UNIX graphics program. Type ‘‘help’’ within gnuplot for more information.



The purview of the above rules may be set using the next set of controls. The calculations may be performed on all pixels within the entire image, just those pixels within the current AOI, or just those pixels which lie outside the current AOI.

Next below are controls which determine the color map to use. Choices include a conventional gray scale, and three different pseudo-color mappings. In addition, any of the color maps may be inverted by using the *Inverse video* control.

At the bottom of this dialog is a graph of the histogram of the pixel values of the image. As with the contrast settings, the pixels being counted may be from the entire image or from just within or without the AOI. The abscissa is the pixel value, and the ordinate is the relative count of pixels with that value. As the mouse roams over the graph the corresponding pixel value and count is displayed. If the *Auto Window* option is active, then the lowest and highest pixels in the graph are the ones which correspond to black and white colors, respectively. Normally, the vertical scale of the histogram is linear, but pressing *Plot log(N)* above the plot changes to a logarithmic scale which will show more detail in areas of the graph with lower counts. Also above the plot are text fields which display the exact pixel values at each end of the graph. These may be changed by typing in a new value and pressing Enter on the keyboard. The end values may also be changed by moving the mouse onto the histogram and pressing the left button; the location of the mouse will determine a new value for the minimum or maximum value to be displayed, depending on whether the mouse is in the left or right half of the graph, respectively. Whenever the histogram limits are changed in either of these ways, they determine a new contrast setting as well and the *Auto Window* option is automatically turned off.



6.6.3 Magnifying Glass and Gaussian fits

This option brings up a dialog box related to the magnifying glass always associated with the cursor. At the top are statistics about the pixels which currently lie within the magnifying glass. These include mean, median, standard deviation, and the value and pixel coordinates of the brightest pixel within the glass. SD>M is the ratio of the brightest pixel over the standard deviation of a local noise annulus.

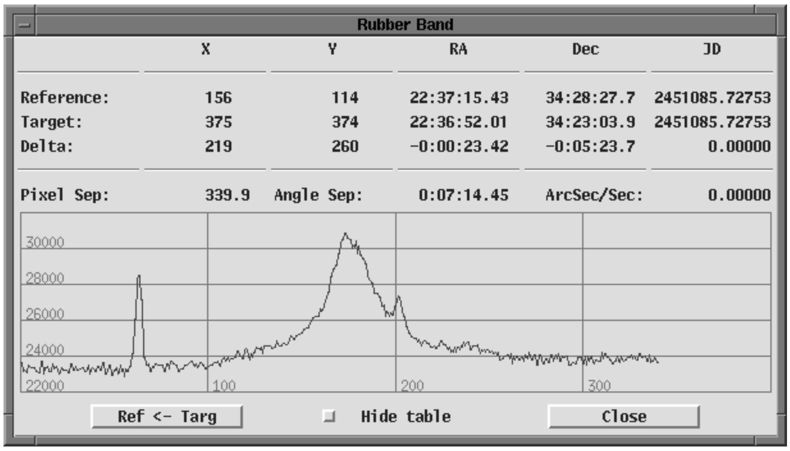
Below the statistics are controls for setting the magnification factor and the size of the magnifying glass box. The size refers to the number of pixels in the original image, not the size after magnification. Note the magnification may be set to 1, which effectively creates a small separate area of interest just for statistics.

Next below is an option *Snap to Max.* When this is on, the center of the magnifying glass is always forced to be over the brightest pixel within a box the size of the glass centered on the cursor. The cursor coordinates are still free to select any desired pixel. To avoid confusion when this dialog is not up, this option is automatically turned off when the dialog is closed.

Next below is an option *Show 1D Plots.* When on, the dialog will display red graphs of the horizontal and vertical cross sections of pixels through the current cursor position (or nearby maximum pixel, if *Snap to Max* is on). The dialog box becomes taller to hold these two graphs. The top graph is along a vertical line centered at the cursor location for the height of the glass box. The bottom graph is along a horizontal line centered at the cursor as wide as the glass. The gray background lines are a relative indication of numerical pixel value and use the same scale on both graphs, although they change dynamically such that the minimum and maximum values along the current cross sections fill the available space in order to show maximum detail.

The last option is to *Overlay Gaussian Fit.* When this is active, the horizontal and vertical cross sections are overlaid with a best-fit Gaussian curve in green. In addition, the center, height and Full-Width-Half-Max values of the Gaussian are displayed in units of pixels. If WCS coordinates are in the image header, the center is also displayed in units if RA and Dec and the FWHM is displayed in arc seconds. To avoid unnecessary computational delays when this dialog is not up, this option is automatically turned off when the dialog is closed.

6.6.4 Rubber band tool



This option brings up a measurement dialog. The top half contains one line for a *Reference* position and one line for a *Target* position. Columns provide the X and Y pixel positions and, if the image header contains valid WCS fields, the RA and Dec positions in epoch 2000. A third line displays the difference in pixel positions and equatorial coordinates between the reference and target positions. A fourth line displays the straight-line separation between the two positions, both in pixels and sky angle.

The bottom half of the dialog displays a cross-section between the reference position and he cursor position. The scale is continually renormalized and nicely labeled to stretch the pixel values encountered along the cross-section to the full height of the available graph space.

To use this feature, move the mouse over the image to a location to be used as a reference and press the left mouse button. If this is the first time this dialog has been up, this location will automatically be made the reference. Otherwise, use the button labeled *Ref<-Targ* to make this location the new reference. Pressing the right mouse button will also make the current cursor position the new reference. Once a reference position has been set, roaming the mouse over the image with the left button down will cause the distance, angle and cross section to be displayed on a continual basis. The reference position is retained if the dialog is closed and later reopened.

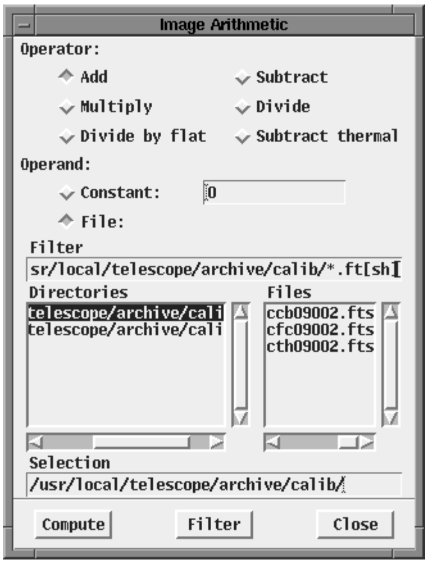
The dialog can also compute a curvilinear velocity. To use this feature, set the reference to a location of, say, an asteroid on one image. Then open an image which contains the same asteroid but taken at a different epoch. Leaving the reference set from the first image, click on the asteroid in the second image. The Julian dates of each image are displayed in the right-most column. These times, In combination with the sky angle separation are used to compute the velocity on the celestial sphere in arc seconds per second.

If only the graph is of interest, screen space may be saved by using the *Hide table* toggle.

An implementation which will allow using three images taken at different epochs to compute the orbital elements of the asteroid is under construction.

6.6.5 Image Arithmetic

This option brings up a dialog which offers the ability to perform simple arithmetic operations on the current image.



The top of the dialog offers choices of what operation to perform. The operations are to add, subtract, multiple or divide. The center of the dialog offers choices of whether the operation will be performed using a numerical value, or another file. If a constant is chosen, enter the value in the text field to the right. If a file is chosen, browse for the file and select it using the file selection box near the bottom. Pressing *Compute* will proceed with the chosen options. In all cases which use files, they must be the same size.

Two additional special cases for file operations are also available as follows.

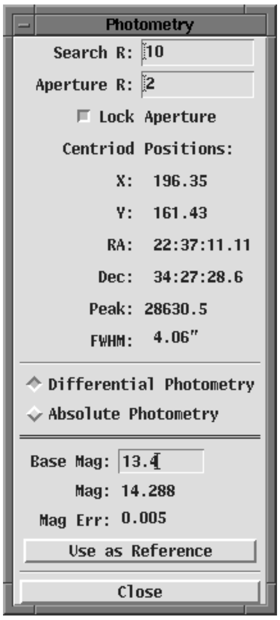
Divide by Flat takes each pixel of the current image and divides it by the value of the pixel at the corresponding location in the operand file, then multiples the result by the overall mean value of the operand file. This is the same arithmetic that is performed by during image correction when applying the flat field correction. The file chosen as the flat must contain the FLATFR field in the header.

Subtract thermal is used to apply a thermal correction file to the current image. This takes each pixel of the current image and subtracts the value of the pixel at the corresponding location in the operand file, then multiples by the ratio of the exposure times of the two files. In this way, the operand file is scaled to allow for different exposure times, in exactly the same way as image correction applies a thermal correction image. Both images must contain the EXPTIME field in the header, and the thermal file must contain the THERMFR field.

6.6.6 Photometry

This option brings up a dialog which may be used to perform either relative or absolute sparse-field aperture photometry. Relative photometry allows the magnitudes of stellar objects to be compared within one image or among several images taken under similar conditions. This method uses an arbitrary star and reference magnitude which may be selected and specified. If the image has WCS headers, use *Options->Label field stars* to display magnitudes from the GSC or USNO SA1.0 catalogs for references.

Absolute color photometry utilizes a set of reference images taken through B, V, R and I filters containing stars of known magnitude which were taken at nearly the same time as the image being investigated. When properly set up, instrument, extinction and color corrections are made automatically and true photometric values will be reported.



The algorithm to determine star energy for both methods uses an aperture approach. The radius of this aperture should be the same for each star in a set for best accuracy. No attempt is made to separate stars whose pixels overlap; hence the restriction that this feature be used only with relatively sparse fields. An implementation which utilizes Gaussian profiles which then can be deconvolved into separate overlapping stars is under consideration.

While this dialog is up, the center mouse button is used to perform photometry. Place the mouse near a star and click to display the information about the star. A red circle is drawn around the star over the image to show its relative magnitude and center for confirmation.

The top of the dialog is common to both modes. At the very top are two fields. The first controls how far to search from the cursor to find the brightest pixel which defines a nominal center for the star. This is useful to avoid having to place the cursor precisely on a star. Larger values are handy for sparse fields; smaller values are necessary for more crowded fields. Below this is an aperture field and a control to lock it. When starting to perform photometry, leave the field unlocked. *Camera* will compute an optimum aperture radius which maximizes the signal-to-noise ratio. Or one may enter a desired radius in the text field. Once a radius is selected, it should be locked for all subsequent comparisons.

Next below is information about the most recent star clicked on. Information includes the location of the center of the star and its peak value. These values are not just based on brightest-pixel but are based on best-fit horizontal and vertical Gaussian profiles and hence are accurate to sub-pixel resolutions. If the image header includes WCS fields, the star location is also displayed in RA and Dec, epoch 2000. The FWHM value is the quadrature sum of the two Full-Width-Half-Max values. This is often a good indication of the seeing disk at the time the image was taken.

Next below is the main choice of whether to perform relative or absolute photometry. The remainder of the dialog depends on which choice is selected.

6.6.6.1 Relative Photometry

To perform relative photometry, choose the *Relative* option in the *Photometry* dialog. The bottom half of the dialog will then present several controls which allow designating a star as a reference and a text field to assign it a magnitude value. As each new stars is selected, its magnitude (with respect to the reference star's magnitude as given in the text box) and an error estimate are presented.

Note that no color correction is applied - this is only a concern if the reference and target star have very different color indices. In any case, it is typically negligible (<0.03 mag) for R and I, but may affect V (up to 0.1 mag) and B (up to 0.5 mag) for B-V differences >1

6.6.6.2 Absolute Photometry

To perform absolute photometry, choose the *Absolute* option in the *Photometry* dialog. The bottom half of the dialog will then present several controls which display the true magnitude of the star through Blue, Visible (yellow), Red and Infrared filters. Selecting the *(Re)Define Reference* button brings up additional dialogs which are used to calibrate the system using the standard Landolt fields.

In order to perform absolute photometry, images must have been taken of standard Landolt fields at about the time of the image to be analyzed. Taking these images can be included easily in a scheduled run by adding them when building the run with *telsched.* Frequent users of absolute photometry soon start to include one or more such standard sets each night as a matter of routine. After the images have been taken, they must be analyzed by *photcal.* This program extracts the coefficients of a model which includes extinction, color correction and instrument calibration. If the images have been taken and processed with photcal, *camera* is all set to perform absolute photometry by just clicking on the star; it’s as easy as relative photometry.

Follow these steps for absolute photometry:

a. For accurate results, always plan to take B, V, R, and I images of the reference star.

b. Load the B image of the reference star first. Select *Absolute Photometry*, then *(Re)Define Reference*. A box will appear in which one enters the known (catalogued) BVRI values of the reference star (in the *True Mag* boxes). Click on the *Apply* button.

c. Depress the *Define Instr Cor* button. This enables Camera to solve for instrumental corrections using the reference star.

d. Load the B image of the reference star, if not already loaded. Click on the reference star with the right mouse button. An instrumental correction constant will appear for the B filter.

e. Repeat step d. for the V, R, and I images of the reference star. All four instrumental corrections have now been determined. Turn off *Define Inst. Cor*.

f. Click *Ok* to remove the reference star dialog box.

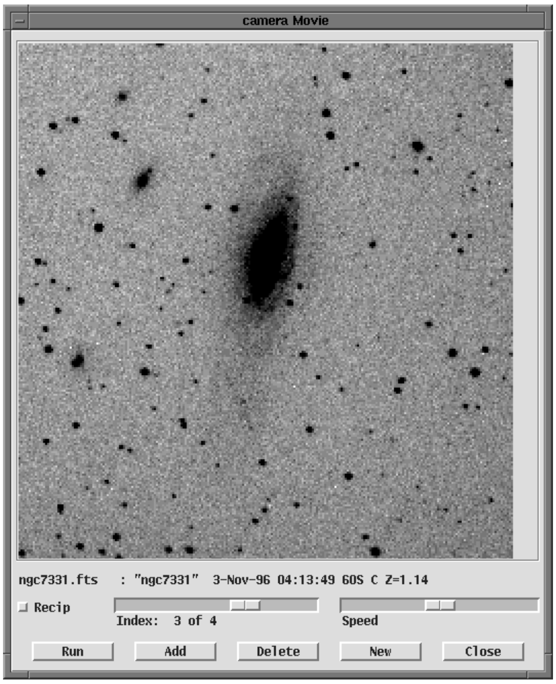
At this point, all photometric results will be computed relative to the reference star results found above. Simply load any image for a given night, click on the target star with the right mouse button, and an magnitude will appear. Camera reads the header information to determine the filter used and the air mass, making suitable corrections.

The only complication is that Camera cannot know the color index of the target star *a priori*. Consequently, Camera displays a column indicating whether a color correction has been applied. This can be done enabled by loading the B and V images of the target star. Camera will recognize that both B and V images have been measured and apply a measured B-V color correction.

See the discussion accompanying the *photcal* command line tool for a description of the photometric model and formats of the Landolt star database and other files which support absolute photometry.

6.6.7 Movie loop

This option brings up a dialog which can hold several screen shots and display them in rapid succession as a movie.



When the dialog is brought up for the first time it will automatically put the portion of the image within the current AOI into the movie as the first frame. Frames may be added after the current index by pressing *Add.* The AOI of the first frame establishes the window reference position and maximum size of all subsequent frames. Only that portion of the image within the current AOI will be used. The current frame may be discarded by pressing *Delete*. All frames may be discarded and a new movie started by pressing *New.*

The movie runs when you press *Run.* The speed of the movie is controlled by the *Speed* slider control. The frames are sequenced from one to max then repeats; or turning on the *Recip* toggle control causes the index to increase to the max then make its way back to frame one in a back-and-forth fashion. The movie may be stopped by pressing *Stop.* The current frame number is shown in the *Index* slider. The frames may be indexed one at a time by clicking in the trough of the *Index* slider. Any particular frame may be displayed by moving this slider to the desired frame number. Using this slider while a movie is running will stop the movie. The status information shown in the main Camera display when the image was added to the movie is displayed beneath the image in the movie window. No image manipulations are possible within the movie window.

6.6.8 Display FITS Header

This option brings up a dialog split into two vertical portions. The top portion displays the FITS header for the current image. The header may be scrolled as necessary both vertically and horizontally.

The bottom portion allows new COMMENT fields to be added to the image header. This only changes the header in memory; be sure to *Save* the image again to make these changes permanent on disk.

The amount of room taken by these two portions may be adjusted by moving the sash which appears between them in the dialog layout. The overall window size may also be changed by adjusting the resize handles in the usual fashion.

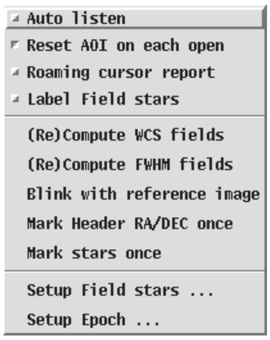
6.7 Other Options

This menu offers several additional features.

6.7.1 Auto listen

This toggle button tells *camera* to listen to a FIFO for the names of images to automatically open and display, each file name terminated with a newline. The name of this FIFO is the value of the X resource *Camera.FifoName*, which is comm/CameraFilename by default, within the $TELHOME directory.

Images being taken during batch scheduled operation and during the autofocus function of *xobs* know to inform *camera* via this method and so it is possible to watch these imaging activities in real time.



6.7.2 Reset AOI on each open

This is an option which, when active, causes the AOI to be reset to 32 pixels inside the border each time a new image is loaded. When this option is off, the AOI remains at the same position when a new image is loaded. If the image is not large enough to contain the AOI, then the AOI will be reset anyway.

Hint: Turning this option off makes it easier to load the movie loop with similar sections of sky from several different images.

6.7.3 Roaming cursor report

This option causes the cursor position and value to be reported across the top of the main window without requiring the left mouse button to be held down.

6.7.4 Label Field stars

This option uses the WCS header fields in the current image to look up and mark each star in the GSC, and possibly the USNO SA1.0, catalog with a blue circle. To the right of each circle is the magnitude from the catalog to .1 magnitude with the decimal point omitted. Note that since the GSC was compiled in the early to mid 1980's, it is often interesting to note the amount to which stars have moved in that time due to proper motion. This feature will not work if there are no WCS headers in the current image. Whether the SA1.0 catalog is used depends on the *Setup Field Stars* option*,* described shortly.

6.7.5 (Re)Compute WCS fields

This option will cause *camera* to match the star-like artifacts in the current image with the GSC, and possibly the USNO SA1.0, and try to find a best-fit pattern match. If successful, new WCS header fields are added to the image. The algorithm is identical to that used by the *wcs* command line utility program. It allows for vertical and horizontal translation and rotation; it does not allow for scaling. Because this can take some time, a separate dialog will appear while the computations are under way which may be used to cancel the effort. Whether the SA1.0 catalog is used depends on the *Setup Field Stars* options*,* described shortly.

6.7.6 (Re)Compute FWHM fields

This option will cause *camera* to compute the median horizontal and vertical Full-Width-Half-Max values of all stars which are not burnt out, which have FWMH > 1 in both dimensions, and which are at least 10 sigma above their local mean noise. The algorithm is identical to that used by the *fwhm* command line tool. Once computed, these values will be included in the header information displayed across the top of the main display.

6.7.7 Blink with reference image

As a special assistance for checking supernovæ and other transient phenomena, this option loads the *movie loop* with the current image and a standard reference image. The OBJECT field in the current FITS header is used to look up the reference image in the special index file $TELHOME/archive/config/blink.idx. The format of this file is as follows:

OBJECT	the value of the OBJECT field to match, no spaces allowed
ERA	The eastern RA of an AOI within the reference image
NDec	The northern Dec of an AOI within the reference image
WRA	The western RA of an AOI within the reference image
SDec	The southern Dec of an AOI within the reference image
filename	name of file containing reference image

Each RA is formatted as HH:MM:SS. Each Dec is formatted as DD:MM:SS.S, optionally preceded with a minus sign.

If the OBJECT name and its reference file are located, the first movie frame is loaded with the reference image, cropped to the AOI indicated in the index file. The second frame is loaded with the same AOI from the image currently being displayed by *Camera*.

6.7.8 Mark Header RA/Dec once

This option causes a small marker to be drawn on the current image based on the values of the RA and DEC FITS header fields. These are each string fields and formatted as H:M:S and D:M:S, respectively. These fields are written based on the telescope position at the time the image was taken. It is generally the nominal center of the image and is used as the seed location for the WCS pattern matching algorithm. If this marker does not fall within the current image, its location is described in the message line at the top of *Camera.* This marker does not survive an image redraw or being covered by anything (hence the qualification "once.").

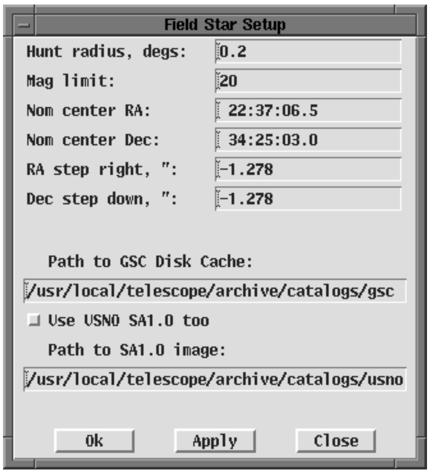
6.7.9 Mark stars once

This option simply searches the current image for all star-like objects and draws a small red square around each. The size of the square has no particular meaning. The star finding algorithm used is identical to that used by all other utilities throughout the OCAAS tool suite. The value of this is to indicate exactly which blobs the algorithm chooses. Challenging tests include random hot pixels, galaxies and burned out stars. The red squares to do not survive an image redraw or being covered by anything (hence the qualification "once.").

6.7.10 Setup Field stars...

This option brings up a dialog which controls how the WCS pattern match is performed and which databases it uses.

The *Hunt radius* is the distance, in degrees, away from the nominal center coordinates which the algorithm will hunt for a pattern match. The *Mag limit* is the dimmest star from the catalogs which will be used. The starting position for the search is specified with the *Nominal center RA* and *Dec* rows of the dialog. These are initialized from the values of the RA and DEC keywords in the FITS header. The pixel scale in each direction is specified by the *RA* and *Dec step right* rows. These are based on the CDELT1 and CDELT2 keywords from the FITS header. These in turn are based on the parameters HPIXSZ, VPIXSZ, RALEFT and DECUP in *camera.cfg.* OCAAS never tries to infer the pixel scale.



The bottom half of this dialog allows setting the paths to the GSC and USNO SA1.0 catalogs. Using the GSC is a minimal requirement and can not be disabled. This dialog also allows choosing whether to use the USNO SA1.0 catalog in addition to the GSC to WCS searches and star labeling. To use the SA1.0, turn on the toggle button and specify the path name to the mounted CDROM. The default path is a symlink which assumes the CDROM has been mounted at /mnt/cdrom. The CDROM is not automatically mounted. Using the SA1.0 is very helpful for images with fields less than about 15 arc minutes on a side because the GSC often does not contain sufficiently many stars for this small field size to form a good WCS solution.

USNO-SA1.0 is a catalog of 54,787,624 sources intended to provide a grid of astrometric reference objects over the whole sky. It is distributed on a CD-ROM free of charge and requests for it should be sent to usno_sa1@sicon.usno.navy.mil.

6.7.10.1 Setting *camera.cfg*

Setting these parameters in *camera.cfg* is an important step in installation. The following procedure can be used, after the telescope is properly tracking and focus has been established so good images are being taken.

In the *Expose->Setup* dialog, turn off both flips and take an image of a known field. Display the same field using XEphem's Sky View, zoomed to approximately match the field of the image. By comparing star patterns in the real image with the Sky View, decide how to rotate the camera so that celestial north will be up in the image, that is, towards increasing Declination. If you are not sure which direction to rotate, just guess and take more images until north is up. Edit *camera.cfg* to make DECUP set to 1.

We now assume the image being displayed has north up. Again using the Sky View as a guide, this time decide whether RA increases or decreases to the left. If it increases, edit *camera.cfg* to make RALEFT set to 1, otherwise set it to 0. We now know how the image hits the camera.

Astronomical images are typically displayed with celestial north up and east left. To follow this convention, edit *camera.cfg* as follows. Set TBFLIP to 0. If you set RALEFT to 1, then set LRFLIP to 0; if you set RALEFT to 0, then set LRFLIP to 1. This completes the orientation issues.

Next we work on pixel scale. Find two stars that are widely separated in the image and the same two stars in the Sky View. Using the *Tools->Rubber band* facility, measure the separation between the two stars on the image in pixels. In the Sky View, do the same by clicking on one star and moving to the next and note their separation in degrees, as reported in the upper left corner of the Sky View. Convert the Sky View separation to seconds and divide this by the separation in pixels from the *Rubber Band.* Then set both HPIXSZ and VPIXSZ to this value. (This assumes your CCD camera has the same number of pixels per unit distance in each direction. If not, choose two stars that are vertically aligned and perform this computation to set VPIXSZ, and use two stars which are horizontally aligned to set HPIXSZ.)

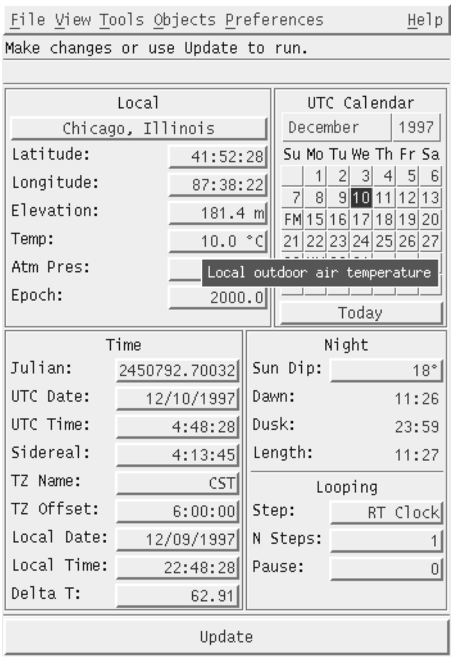
For example, if Sky View reports the separation is 0:9:17, and *Rubber Band* says the separation is 478 pixels, HPIXSZ and VPIXSZ should be set to (9x60 + 17)/478 = 1.165.

Save the new *camera.cfg* and restart camera.

6.7.11 Setup Epoch...

This brings up a small dialog which allows the display epoch used by *Camera* to be specified. This applies to the RA and Dec values which are displayed in the top message line while the cursor is roaming over an image and positions computed during photometry.

7 XEphem -- ephemeris planning and mapping tool



XEphem provides a rich set of astrometric calculations and attractive graphical and image displays in an easy-to-use GUI framework. It can also be used within the OCAAS system to display and directly control the pointing direction of the telescope. *XEphem* includes extensive on-line context-sensitive help and real-time help tips on all controls. That level of detail is not repeated in this document, but rather we present only the highlights of what is possible.

XEphem can compute information on demand or time can be set to increment automatically. In this way a series of computations and movies can be generated unattended.

RA/Dec calculations may be topocentric or geocentric, and apparent or astrometric. When the Epoch is set to a fixed date the values are astrometric, that is, corrected only for precession and light travel time. When the Epoch is set for EOD (Epoch of Date) the values are apparent and are also corrected for nutation, aberration and deflection. Topocentric values are also corrected for parallax. All Alt/Az values are always topocentric and are corrected for refraction.

7.1 Command line arguments

XEphem supports the following command line arguments.

-help prints a summary of command line arguments and exits.

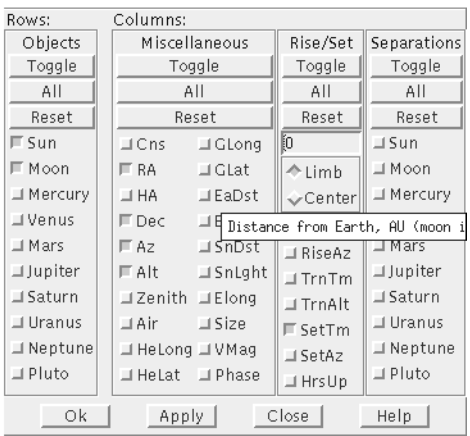
-prfb prints the default list of fallback resources and exits.

-install can be followed with either *yes, no* or *guess.* The first two directly control whether *XEphem* will create, install and use a private colormap. *Guess* tells it to decide automatically. A private colormap is only needed if the X Server does not support enough colors.

7.2 Primary display capabilities

Follows are descriptions of the graphical displays supported by *XEphem*. All displays can be printed or saved as Postscript files at any time.

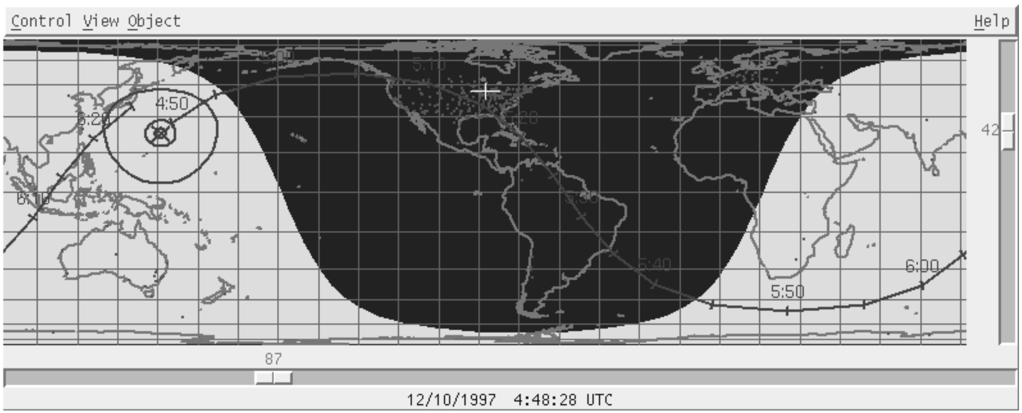
7.2.1 Configurable Data table



The Data Table is a matrix with one row per object and columns of various quantitative information. The rows can be set to any of the objects in the *XEphem* database. The columns can be set to include items including apparent or astrometric RA and Dec, local azimuth and altitude, true heliocentric coordinates, distance from sun and earth, light travel times, galactic coordinates, solar elongation, angular size, visual magnitude, illumination percentage, local rise and set times and azimuths, local transit times and altitude, length of time up, constellation, and angular separations between all combinations of objects. The desired rows and columns are selecting using the *Control->Setup* dialog shown here.

7.2.2 Earth view

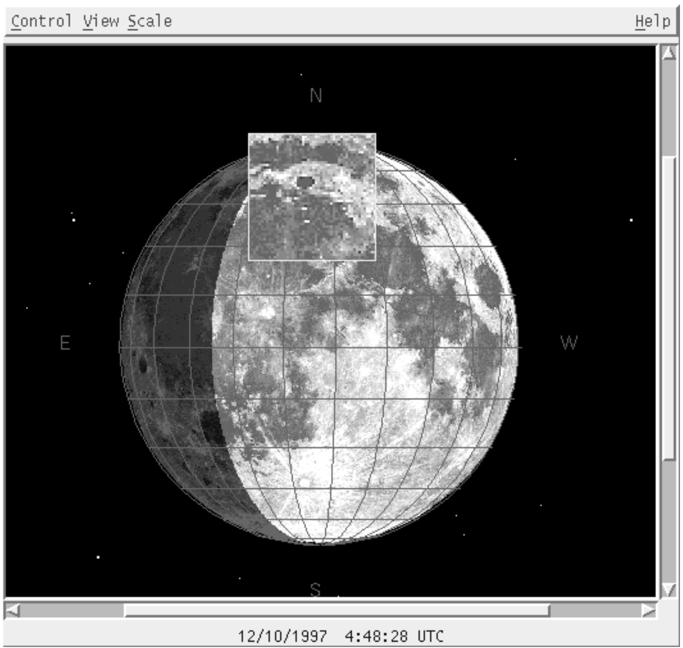
The Earth view displays a simple landmass map of the Earth surface in either a spherical or cylindrical projection. The latter allows seeing the entire surface of the earth at one time. The map can be overlaid with several items including the area currently in sunlight, locations of major cities and observatories, latitude and longitude grid, subearth location of any object in the *XEphem* database, and totality location if a Solar eclipse is in progress. The spherical projection can be oriented so the user appears to be in space looking down over any desired location. The locus of points at which any object is 0, 30, 60 and 90 degrees above the horizon is displayed around each subearth point. The ground track of objects over time may be displayed, which is particularly suitable for artificial earth satellites.



Mouse interactions within the Earth view include a roaming display of Local Sidereal Time, Latitude, Longitude, and Local Mean Time while the left button is depressed. If the right mouse button is clicked over an arbitrary location the latitude and longitude is displayed as well as the local altitude and azimuth of the current object and, if the object is an artificial earth satellite, the range and radial velocity of the object. If a location is selected which is near a city or observatory its name, location and local mean and sidereal are also displayed.

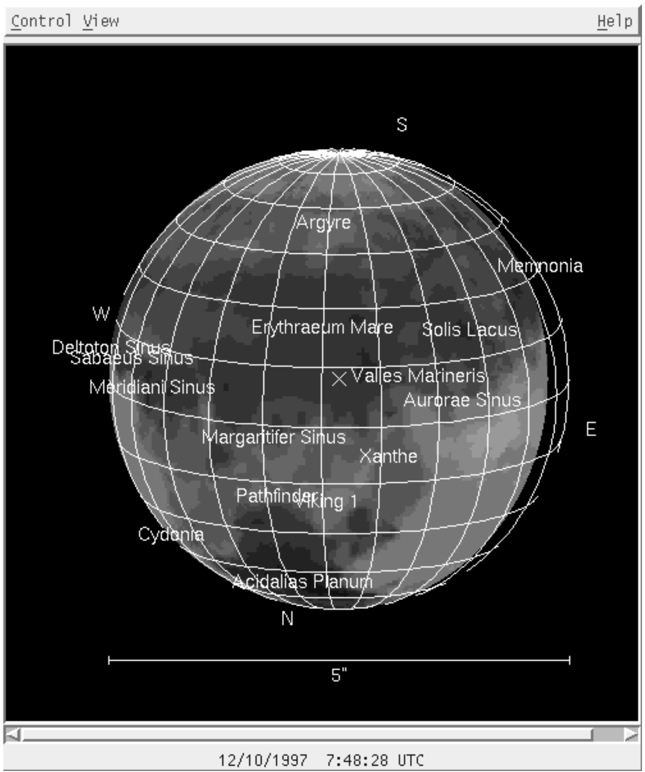
7.2.3 Moon view

The Moon view displays a detailed gray scale image of the front side of the moon, with the current portion in shadow displayed somewhat darker. The shadow brightness can be controlled with the *Earthshine* control from the *Control* menu. The sky behind the moon can be left dark, or it can be populated with objects from the current *XEphem* database, including the Hubble GSC and SAO, HD and PPM catalogs. The map may be flipped horizontally and vertically. During a lunar eclipse, the umbra and penumbra boundaries may be displayed. The view may be labeled from a database of several thousand features, including landing sites of several spacecraft. The size of the image can be set to several values and panned if larger than the current window. As the size increases more features are be labeled. A dot on the limb of the image shows the point which is tilted most towards the Earth due to libration. A grid at each 15° interval of longitude and latitude may be laid over the image. Also when the grid is enabled the current sub-Earth and sub-Solar positions are marked.



As the mouse moves over the Moon view with the left button depressed, a magnifying glass shows the lunar surface magnified by 2x in width and height. Depressing the right mouse button will display the name, type and approximate size of the lunar feature nearest the cursor and the solar altitude at that location. If the right mouse button is depressed over an object in the sky background, its name and magnitude are displayed and a shortcut is provided to assign the object one of the user defined objects and hence add it as a row in the Data table.

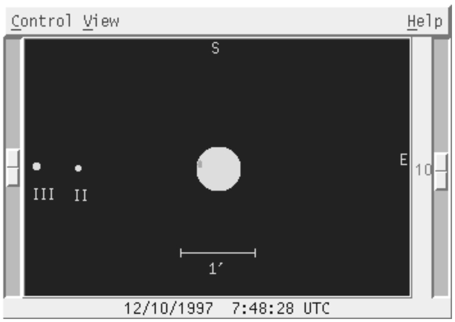
Additional quantitative information is available using the *View->More Info* dialog. This always displays the current tilt and libration in longitude and latitude, and the current sunrise longitude and subsolar latitude. If the left mouse is depressed, it also displays the selenographic latitude and longitude of the cursor, and the times of the next sunrise and sunset at the cursor location.



7.2.4 Mars view

The Mars view displays an image of Mars morphed onto a sphere and oriented to match the current *XEphem* date and time. The image is derived from Voyager spacecraft data taken at 1° resolution. A grid at each 15° interval of longitude and latitude may be laid over the image. Also when the grid is enabled the current sub-Earth position is marked. The image may be flipped horizontally and vertically, and rotated to place any location in the center by entering coordinates or using the mouse to indicate a new center position. If a value for the local seeing is entered in arc seconds, the image will be artificially blurred based on the current disk size and image scale to simulate the view under those conditions through a telescope. A small database of natural features and spacecraft landing sites can be enabled to display labels on the map. The sky behind the map can be left dark, or it can be populated with objects from the current *XEphem* database, including the Hubble GSC and SAO, HD and PPM catalogs.

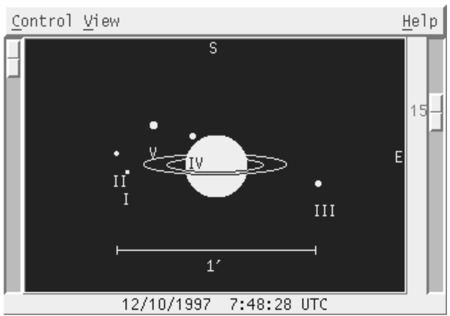
7.2.5 Jupiter view



The Jupiter view displays a simple schematic graphic of the planet, the Great Red Spot and the four major Galilean moons. The display may be flipped horizontally and vertically. The sky behind the display can be left dark, or it can be populated with objects from the current *XEphem* database, including the Hubble GSC and SAO, HD and PPM catalogs. Quantitative information about Jupiter and its moons are also available in a separate dialog by using the *View->More Info* option. Data displayed include the current Central Meridian Longitude in Systems I and II, and the RA, Dec and visual magnitude of Jupiter and each moon. A simple Movie Demo mode is available to watch the moons orbit.

The right mouse button may be used to identify any moon or sky background object, and a shortcut allows the latter to be immediately assigned to one of the ObjX/Y/Z user objects (and thereby added to the Data table as a new row).

7.2.6 Saturn view

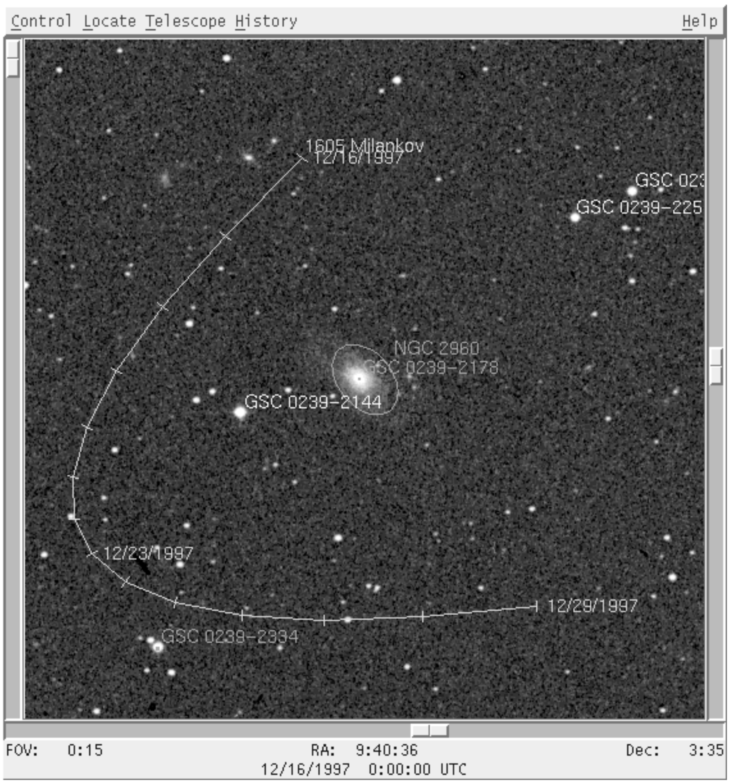


The Saturn view displays a simple schematic graphic of the planet, the rings and the eight major moons. The display may be flipped horizontally and vertically. The sky behind the display can be left dark, or it can be populated with objects from the current *XEphem* database, including the Hubble GSC and SAO, HD and PPM catalogs. Quantitative information about Saturn and its moons are also available in a separate dialog by using the *View->More Info* option. Data displayed include the current ring tilt as seen from the Earth and Sun, and the RA, Dec and visual magnitude of Saturn and each moon. A simple Movie Demo mode is available to watch the moons orbit.

The right mouse button may be used to identify any moon or sky background object, and a shortcut allows the latter to be immediately assigned to one of the ObjX/Y/Z user objects (and thereby added to the Data table as a new row).

7.2.7 Sky view

The Sky View is a major workhorse of *XEphem* with many features. It can display sky maps of all objects at scales from 180° full-sky down to 5 arc minutes along each side. The orientation may be Alt-Az or RA-Dec.



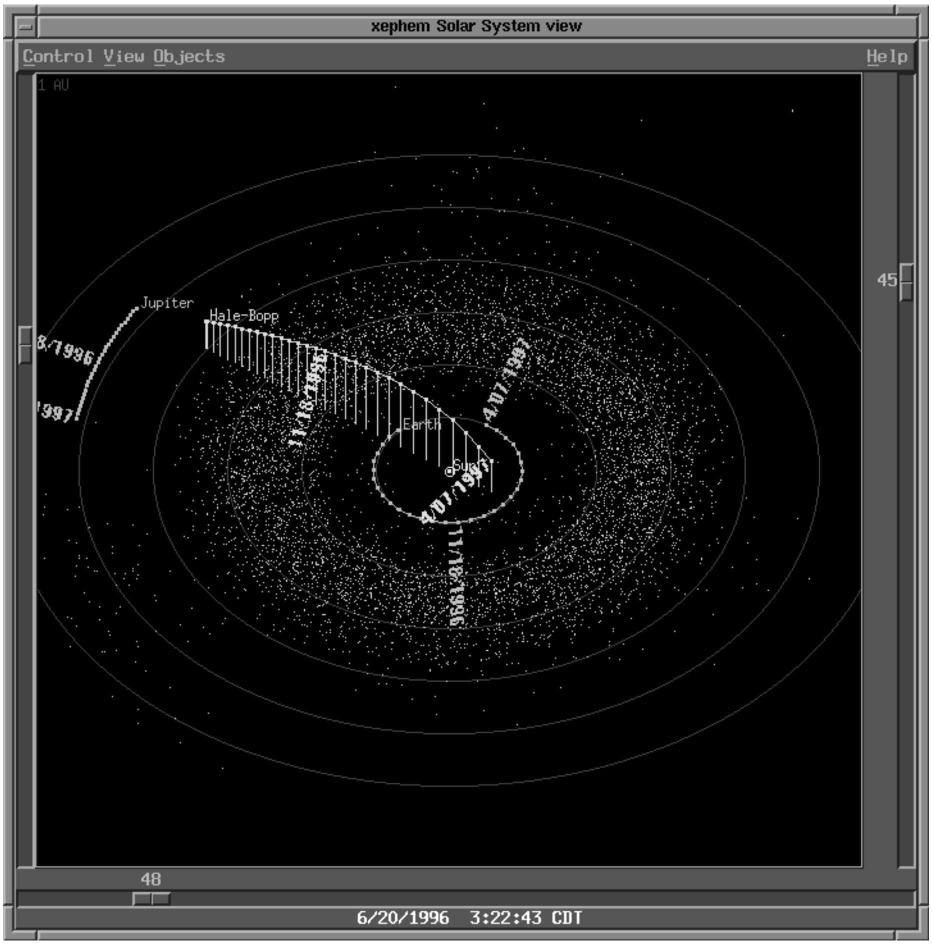
Many options allow fine control of names and labeling; separate deep-sky and stellar cutoff limiting magnitudes; a coordinate grid; equatorial, ecliptic and galactic planes; constellation names, boundaries and figures; filter objects by type and magnitude; display time-sequenced trails of any number of objects simultaneously; measure separations; overlay round or square eyepieces of adjustable size; quickly locate the planets or any ObjX/Y/Z user object; save the current option settings in a history list and easily reinstate them later in the session; and more.

The Sky View can display FITS files of images and overlay them with all graphical information if they include WCS header fields. If connected to the Internet (firewalls and SOCKS proxy servers are supported), images may also be easily pulled from the Digitized Sky Survey maintained on-line by STScI and ESO.

If the left mouse button is depressed while roaming over the sky view, the RA, Dec, Alt, Az, constellation and separation from the initial mouse-down position will be displayed in the corners of the window. The right button brings up a popup menu containing basic data about the nearest object, including name, RA, Dec, altitude, azimuth; size; magnitude; spectral class; and optionally, times of rise, transit and set. The popup also includes many options including control of center and zoom in and out; place an eyepiece; designate an object to be ''tracked' which means keep it centered as *XEphem* time is advanced; assign an object to one of the ObjX/Y/Z user objects (and thus immediately add it as an active row in the Data table); reload GSC and PPM catalog objects; and more.

The Sky View is also an integral part of the OCAAS telescope command system. The current position of the telescope may be labeled on the map in real-time. And the telescope may be slewed to any location or object and begin tracking by just pointing to the object and selecting *Set telescope* from the popup menu. These features enabled via the Telescope menu.

7.2.8 Solar System view



The Solar system view shows a perspective drawing of the solar system. Included are all planets plus, optionally, all solar system objects currently loaded in the *XEphem* database such as comets and asteroids. The eye position may be rotated in heliocentric longitude, tilted above and below the ecliptic in latitude, and zoomed in and out from the Sun (which always remains at the center of the window). Time-sequenced trails may be easily defined for all objects to show their path through the solar system. Objects can be drawn with a line connecting them to the ecliptic to help show their true location in space. A second window may be displayed beside the first which will show the solar system in 3-Dimensions; both crossed-eyed and relaxed-infinity visual fusing methods are supported.

Selecting an object with the right mouse button displays basic data including its RA, Dec, magnitude, Earth and solar distance and heliocentric coordinates. A shortcut is also provided to easily add the object to a new row in the Data table.

7.3 Research tools

As interesting and entertaining as the visual displays are in *XEphem,* perhaps its greatest value is its quantitative features. The algorithms used for the planets are directly traceable to the JPL DE200 numerical integration and will match the *Astronomical Almanac* to full displayed precision.

The natural satellites models were supplied by the French Bureau of Longitudes and are accurate to 0.5 arcsecond between the years 1996 and 2020 (2Q98) . Topocentric apparent place is computed by starting with astrometric mean place (including Solar deflection) and applying corrections for precession, nutation, aberration, parallax, and refraction (the latter two depend on accurate values for latitude, longitude, elevation and atmospheric pressure and temperature). Comet and asteroid ephemerides do not include perturbations but timely orbital elements are readily available for downloading via the Internet. Many sites support *XEphem's .edb* file format directly such as http://cfa-www.harvard.edu/iau/info/OpticalObs.html, plus OCAAS includes *perl* and *awk* converters for many popular web site formats such as the asteroids database at ftp://ftp.lowell.edu/pub/elgb/astorb.html.

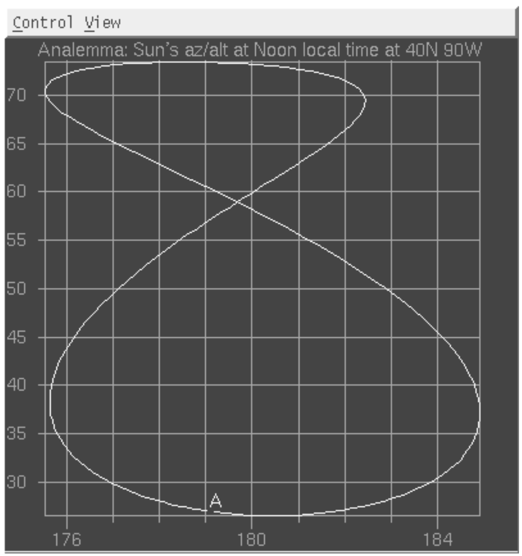
All of this accuracy can be tapped directly in several ways, as described in the next several sections. Many of these features take advantage of one common trait: all data fields throughout *XEphem* may be turned into Pushbuttons which allow them to be easily registered for use by these features. This eliminates the need to make manual notes or perform data copy-paste, and allows the data to be available programmatically at various date and time settings.

7.3.1 Solve user-defined equations of circumstances

One or more data fields may be combined into a user-defined arithmetic or logical function. Operators include all common arithmetic, boolean, trig and log functions, as well as square root, absolute value and more. This function may then be compiled and ''solved'' for the time at which it is a local minimum or maximum, when it goes to 0, or when its boolean value changes state.

This feature may also be used to define any new field derived from any other fields. This is because the user function is always computed and displayed (and is also always available for other uses such as plotting or listing) whether or not it is currently being ''solved.'

User functions allow a limitless range of astrometric questions to be investigated. For example, times of occultations; orbital extrema; and combined constrained effects may be discovered even though they are not directly supported by *XEphem.*



7.3.2 Create user-defined plots of any data

As many as ten pairs of data fields may be stored as they are computed and later displayed as 2-D plots. The format of the file is simple ASCII text suitable for importing by programs such as *gnuplot. XEphem* also includes its own basic plotting capability.

7.3.3 Export any data to a text file

Any of the data fields displayed may be written to a simple ASCII text file as they are computed. The file is formatted into columns, with optional headers, for easy reading directly or may be imported into word processing or other programs.

7.3.4 Sort all pairs of objects by increasing angular separation

All pairs of objects currently in the *XEphem* memory database (but not the GSC and PPM Field stars) may be sorted in to increasing separation with essentially one menu selection. This is designed to search for appulses or other interesting or esthetic arrangements of objects.

7.4 Other features

Follows are more highlights of *XEphem*.

7.4.1 Display FITS and DSS images with graphical overlays

As described in the Sky View, FITS files may be displayed overlaid with object names and other graphical annotations (if they include WCS header fields). If an Internet connection is available, the images may be retrieved and saved locally from the Digitized Sky Survey.

7.4.2 Astronomical catalogs and object management

XEphem uses its own database file format. It is well documented and is a simple ASCII text format which the user may easily utilize for their own collections. The program has controls for reading these files into memory; checkpointing collections of objects; deleting all objects or back the last checkpoint; displaying statistics as to the number of each type of object loaded; and creating new objects in memory on the fly. Up to three of these objects may be defined. They are referred to as user defined objects and are denoted as ObjX, ObjY and ObjZ. These objects serve two roles. They may indeed be new objects. Or they may refer to objects which were actually loaded from a file but by assigning them to a user defined object they can be accessed via several shortcuts throughout *XEphem.*

Many standard deep sky catalogs come with *XEphem* (and hence OCAAS*)* including Messier, RNGC, IC, UGC, SAC, and Abell. Stellar catalogs include YBS and GCVS. Also included are VLA radio and IRAS IR reference objects. The Minor Planet Center and Center for Astronomical Telegrams issue comet and other time critical phenomena reports on the Web directly in *XEphem* format. The frequent posting of asteroid elements in the current epoch by Lowell Observatory are supported with a small *perl* conversion utility program, supplied.

The Hubble GSC, SAO, HD and PPM catalogs are also supported in specially formatted files for compactness and speed. The PPM is the Position and Proper Motion catalog which allows for display of nearby stars corrected for their proper motion of several thousand years.

8 Command line tools

The following programs are available from a UNIX command line, such as from an *xterm* window. All programs print usage summaries and exit if executed with the *-help* option or with any unidentified option.

These programs are written in this form primarily so they may be used in higher level specialized scripts to perform automated extensive processing on large numbers of files. Indeed, OCAAS uses several of them in just this way during batch scheduled processing. They are also useful for calling from within cgi-bin web scripts.

GUI wrappers around these programs may be available in later releases, depending on demand.

8.1 Image Manipulation Tools

8.1.1 *calimage* -- calibrate a raw image file

This program can take one or more raw images on the command line and applies appropriate Bias, Thermal and Flat correction files. The results are stored over the original file *in place* so copy the files first if you want to keep the raw images.

By default, the correction files are searched for in the directory $TELHOME/archive/calib. Calimage accepts three command line arguments, -b, -t, and -f, which can be followed by a specific file name to use in lieu of the standard bias, thermal and flat file, respectively.

This program can also create new calibration files from raw images. Bias files must be taken with zero duration and the shutter closed. Thermals must be taken with the shutter closed. If several files are given they are averaged. All FITS keywords appropriate to the correction file type being created are added.

See the chapter on *camera* for a full discussion on correction images.

Usage:

calimage: {-C|-B|-T|-F} [options] *.fts

purpose: to apply or create image calibration correction files.

Exactly one of the following options is required:

-C	fully correct each given raw file in-place
-B	create a new bias corr file from the given set of raw biases
-T	create a new thermal corr file from the given set of raw thermals
-F	create a new flat corr file from the given set of raw flats (see -l)

The following options may also be used to specify the use of specific correction files, a different catalog directory, or other features:

-b fn	specify a specific bias correction file, fn

-t fn	specify a specific thermal correction file, fn
-f fn	specify a specific flat correction file, fn
-d dn	specify an alternate catalog directory, dn
-l f	specify the filter, f, for -F if not in header info

8.1.2 *crop* -- reduce image sizes

Crop is a program which crops a set of FITS images to the largest common AOI. Or, crop can be given a text file which lists OBJECT names and AOIs and it will crop each image to that AOI. A typical such file is the one used by *camera* for blinking supernovæ candidates, *blink.idx* located in $TELHOME/archive/config. Or, crop can be given an AOI on its command line and it will crop all images to that such.

N.B. It creates the new cropped images *in place*. Copy the images first if you want them preserved.

Usage:

crop [options] [*.fts]

-a AOI	crops all images to the given AOI. The AOI is in the form:
	"East_RA North_Dec West_RA South_Dec", RA as H:M:S, Dec as D:M:S
-c	crops all images to their maximum common AOI.
-l file	crops to AOI for matching OBJECTs listed in given file. each line in file format is:
	OBJECT East_RA North_Dec West_RA South_Dec.
-f file	list of files to crop.
-v	verbose

8.1.3 *fcompress* -- compress an image file

This tool compresses a FITS file using the H-Compress algorithm developed at the Space Telescope Science Institute. The algorihm can be lossless in which case it will generally achieve between 2x and 3x reduction in file size. Lossy compression is specially tailored to discard only noise in the sky background which is not quantitatively interesting. Scale factors up to 100x can often still be used for photometric or astrometric uses. At such extremes, one will notice that the sky background becomes too patchy for ‘‘pretty pictures’’ but the stars remain well defined Gaussian forms.

The OCAAS file name extension for uncompressed FITS images is .fts. Files compressed with this algorithm are created with the extension .fth.

Fcompress compresses only the pixel data in a file and leaves the FITS header unchanged. It does, however, add two new fields to the header. HCOMPSCAL is an integer field which records the scale at which the image was compressed. HCOMSTAT is a string field which records the status of the compression. It will contain the string ‘HCOMPRSD’ if the pixels are compressed in this file, and the string ‘UNHCOMP’ if they have been recreated by *fdecompress*.

Usage: fcompress [options] [*.fts]

-f file	name of file containing names of files to compress. Files must either be specified on the command line or with this option, but not both.
-s scale	specifies the scale factor for the compress algorithm. Larger values indicate greater compression but this is *not* directly related to the expected file size reduction. A value of 1 indicates lossless compression. The default is 666.
-r	after creating each .fth compressed file, delete the original .fts file.

8.1.4 *fdecompress* -- uncompress a compressed image file

This tool decompresses FITS file compressed using *fcompress.* It sets a FITS string header field HCOMSTAT to ‘UNHCOMP’ when successful.

Usage: fdecompress [options] [*.fth]

-f file	name of file containing names of files to decompress. Files must either be specified on the command line or with this option, but not both.
-r	after creating each .fts uncompressed file, delete the .fth compressed file.

8.1.5 *findstars* -- find all star-like objects

This tool finds all star-like objects in the given FITS files and prints their locations. It is the same algorithm used by the option *Mark stars once* in *camera.*

Usage:

findstars: [options] *.fts

-r	report in rads -- default is H:M:S D:M:S

8.1.6 *fitsfilter* -- various image processing filters

This tools performs various image processing filters on the given FITS files, *in place.* Additional filters will be added over time. A HISTORY field is added to the header to record the processing performed.

Usage:

fitsfilter: usage: [options] file.fts ...

Purpose: perform a filter on given fits files IN PLACE.

-f file	name of file containing filenames, one per line.
-l n	apply order-n 2d polynomial flat-field
-m s	median filter of size s

8.1.7 *fitshdr* -- display and edit FITS header fields

This tool displays or edits the FITS headers of image files. All editing is performed *in place* so copy the file(s) before editing them if the original version is to be retained.

Usage: [options] [*.fts]

-f file	name of file containing names of files to edit. Files must either be specified on the command line or with this option, but not both.
-d name	delete all fields *of any type* with this name
-i name val	add (or replace) an integer field of the given name with the given value
-l name val	add (or replace) a logical field of the given name with the given value, which must be specified here as 0 or 1 (which will be turned into F or T in the actual header)
-r name val	add (or replace) a real (floating point) field of the given name with the given value
-s name val	add (or replace) a string field of the given name with the given value. Be sure to surround the value with quotes to if it contains spaces or other characters with special meaning to your shell.
-c comment	add a COMMENT field. Be sure to surround the comment with quotes to if it contains spaces or other characters with special meaning to your shell.
-p	just print the FITS header
-P	same as -p but prefixes each line with the file name

As many options for each type and operation may be specified as desired.

8.1.8 *flipfits* -- flip rows or columns

This program flips FITS image file rows and/or columns. It only moves the pixels, it does *not* modify the existing headers but it does add HISTORY entries denoting the changes.

usage: [options] file.fts ...

-r: flip rows.

-c: flip columns.

N.B. It creates the new flipped images *in place*. Copy the images first if you want them preserved.

8.1.9 *FWHM -- add FWHM fields to header*

This program finds the median horizontal and vertical full-width-half-max star statistics for an image and records them in the header. The algorithm uses only up to the 20 brightest stars in the image whose brightest pixel is no larger than 60,000 (to eliminate burned out stars), have SNR at least 10:1 (to avoid overly dim stars), a FWHM in each direction greater than 1 (to avoid random hot pixels), and which lie inside a border of 32 pixels in from each edge (to ignore typical chip artifacts).

The FITS fields created are defined as follows:

FWHMH	Horizontal FWHM median, pixels
FWHMV	Vertical FWHM median, pixels
FWHMHS	Horizontal FWHM standard deviation, pixels
FWHMVS	Vertical FWHM standard deviation, pixels

usage: [options] [file.fts ...]

-a	print in arc seconds if can, else pixels (requires CDELT1 and CDELT2 FITS fields)
-v	verbose
-o	allow overwriting any existing FWHM fields
-w	write header (default is read-only)
-d	delete any FWHM header fields (requires -w)
-f file	name of file containing filenames, one per line.

8.1.10 *pterrors* -- compute pointing error mesh

This program is used to analyze a collection of images and produce a map of pointing errors which span the area of the sky covered by the images. The images need not be regularly spaced. The images must already have WCS header fields (see *wcs)*.

The output file format is in four columns. The first two columns define a sky position, and the second two are the errors in each axis encountered at that position. Options allow the coordinate system reported to be either equatorial HA/Dec or horizon Alt/Az. This format is easy to use with most plotting programs to help visualize and analyse the nature of the errors. See the Installation chapter for an example use *gnuplot.*

The default output format of the program happens to also be formatted so it can be used directly as the telescoped.mesh file used by *telescoped.* This file resides in $TELHOME/archive/config by default. Telescoped will automatically use it if it can read it when it starts. *All program defaults are correct for creating the telescoped.mesh file for use with telescoped -- use no options.*

Usage:

-a	print in Alt/Az format
-c	multiply HA by cos(Dec) (or Az by cos(Alt) if -a) to form sky angle
-p	prefix each line with filename
-t	print telescope axes at each HA/Dec
-v	verbose

HA/Dec Output format (default):

Target HA, hours

Target Dec, degrees

Target HA - WCS HA, arcminutes (*cos(Dec) if -c)

Target Dec - WCS Dec, arcminutes

Alt/Az Output format (-a):

Target Az, degrees

Target Alt, degrees

Target Az - WCS Az, arcminutes (*cos(Alt) if -c)

Target Alt - WCS Alt, arcminutes

Telescope Axes Output format (-t):

WCS HA, rads

WCS Dec, rads

RAWHENC, rads from home

RAWDENC, rads from home

8.1.11 *ptmeshadd* -- add two pointing meshes

This program is intended to be used when updating the pointing mesh used by *telescoped*. A typical scenario is to perform a basic axis calibration, then take a mesh of images to map the pointing errors and use this with telescoped. The computations to include the WCS headers in those images can be somewhat lengthy to perform because the pointing error may be fairly large at odd places in the sky. When the mount is suspected of being moved slightly, or just to be sure the corrections are as good as they can be, it is often tempting to rerun the mesh from time to time. Rather than do a mesh from scratch again, that is, remove the telescoped.mesh file and start with just the basic mount mode, it can be more efficient to take a new pointing mesh while the previous one is still in place because the WCS solutions will probably go much faster. The resulting mesh will in effect be a map of the errors in the mesh. To use this new second mesh directly would be incorrect because it does not include the errors already mapped by the first mesh.

To allow the second mesh to be used, this program will add the errors in the first mesh to the errors in the second mesh to produce a third mesh which is the composite of the two. Installing this composite mesh in place of the old telescoped.mesh will utilize the information captured in both meshes. Pterrors does not require that the two meshes coincide or even be performed at the same scale. Given two meshes to add, pterrors will interpolate the errors found in the second file to coincide with the locations of the errors in the first file. The output will have points at the same positions as found in the first file. Thus, for maximum effectiveness, the first file should be the denser of the two meshes. The interpolation radius should be at least twice the spacing of the sparser mesh.

Usage:

ptmeshadd: [-options] file1 file2

Add pterrors reports file1 and file2 at file1 points.

-v	verbose
-r n	max interpolation radius, degrees; default = 10

8.1.12 *wcs* -- add World Coordinate System fields to header

This tool takes one or more FITS images and sets the traditional WCS header fields by matching star patterns in the GSC and/or the USNO SA1.0 catalogs. The algorihm begins at the scale and location specified by the following four FITS header fields, which must already exist:

RA	nominal RA of field center; string type; format ‘‘HH:MM:SS.S’
DEC	nominal Declination of field center; string type; format ‘‘ DD:MM:SS.S’
CDELT1	horizontal pixel scale, RA step right, real type, degrees/pixel
CDELT2	vertical pixel scale, Dec step down, real type, degrees/pixel

If these fields exist, then *wcs* extracts all star-like objects from the image and starts a spiral sweep comparing these patterns to entries in the GSC. The spiral steps by 1/3 image width or height each time up to a maximum specified by the -u option. If a match is found, the remaining standard WCS header fields, listed below, are added to the header. If no match is found, the header is unchanged. The program *wcs* exits with a value of 0 if all files were successfully fit, or with an integer indicating the number of images for which no fit was found.

The traditional WCS FITS fields which are added if a fit is found are as follows:

CTYPE1	RA---TAN
CRVAL1	RA at CRPIX1, degrees
CRPIX1	RA reference pixel index, 1-based
CROTA1	0.0
CTYPE2	DEC--TAN
CRVAL2	Dec at CDPIX2, degrees
CRPIX2	Dec reference pixel index, 1-based
CROTA2	Rotation N through E, degrees

Usage:

wcs: usage: [options] [file.fts ...]

-r dir	use GSC from cdrom at dir (default is off)
-c dir	alternate GSC cache directory (default is archive/catalogs/gsc)
-n dir	alternate USNO directory (default is archive/catalogs/usno)
-v	verbose
-o	allow overwriting any existing WCS fields
-w	write header (default is read-only)
-d	delete any WCS header fields (requires -w)
-u rad	hunt radius, degrees (default is 1)
-f file	name of file containing filenames.

8.2 Research Tools

The following programs are available from a UNIX command line, such as from an *xterm* window. All programs print usage summaries and exit if executed with the -help option or with any unidentified option.

These tools are intended to be a powerful adjunct to research efforts, particularly those involving time varying phenomenon.

8.2.1 fphotom -- absolute field photometry

Fphotom is a program for producing true photometric magnitudes for all stars in a color field, that is, all stars that are found in common among a set of four images taken through B, V, R and I filters. Photometric calibration constants for the night of the image set are read from the file photcal.out, or from a set of standard defaults in photcal.def if no date matches. These files are in $TELHOME/archive/photcal by default.

The output is a table, one star per line, of star positions, their B, V, R and I values, with error estimates, and the geometric mean and ratio (Y to X) of their full-width-half-max values.

Usage:

fphotom: [options] config_file

Options:

-c file	photometric constants file. default is photcal.out
-d file	default file to use when date not found. default is photcal.def.
-p file	directory of out/def files; default is /usr/local/telescope/archive/photcal
-v	verbose.

Required:

config_file

The config_file is required and must consist of exactly either 2 or 5 lines. In either case, the first line contains two fields, separated by whitespace, to define the photometry parameters, as follows:

rsrch	max radius to search for brightest pixel
rap	aperture radius, pixels.

The file may then have four additional lines which are the names of images of the same field through B, V, R, and I filters. The filenames may be given in any order.

Or, the file may have one additional line which begins with a * and is the name of a file which contains the names of the four BVRI filter images, exactly as though they appeared directly in the config file as above.

8.2.2 photcal -- produce calibration constants for absolute photometry

Photcal scans a directory of images of Landolt photometric standard stars and, along with standard star name information in photcal.ref and default k'' values from photcal.def, determines a set of 8 photometric parameters: V0 and k' for each of 4 colors, BVRI, which best-fits the image data. These values are written in a form suitable for use with *fphotom* and *camera* so, by default, photcal appends the new set of values with a date header to the file photcal.out. The values of k'' are held constant and may be read from photcal.def or changed on the command line. The files all live in $TELHOME/archive/photcal by default.

The photometric model is from Simon et al 1994[*] and can be summarized as follows:

---

[*] *Measuring Filter Response and Extinction Coefficients Using CCD Observations of Photometric Standard Stars,* Leslie Simon and Robert Mutel, I.A.P.P.P. Communications No 57, p48-53, Autumn 1994

$$V = V_{obs} + V_0 + k'Z + k''(B-V)$$

where:

V	true apparent magnitude	per star, from photcal.ref
V_{obs}	observed magnitude	per star, from pixels
V_0	instrumental correction	to be solved
Z	air mass	per file, from header
B-V	color index	per star, derived from photcal.ref
k'	airmass dependency	to be solved
k''	color index dependency	to be solved

This equation is repeated for each color B, V, R and I. Thus, the total solution consists of 8 numbers: V_0 and k' for each color. (Be careful not to confuse the color V with the V in the equation).

Stars with SNR < 4.0 are rejected.

Usage summary;

photcal: [options] mm/dd/yyyy

Options:

-i dir:	directory of images; default is $TELHOME/user/images.
-p file:	directory for def/ref/out files; default is $TELHOME/archive/photcal
-d file:	default k'' values filename; default is photcal.def
-r file:	field ids filename; default is photcal.ref
-c file:	calibration filename to append to (- means stdout); default is photcal.out
-k B V R I:	values of k'' for each color; may be set from file using -d or built-in defaults are: -0.46 -0.15 -0.02 -0.06
-t days:	+/- days image's JD may vary from given date; default is 0.5
-v:	verbose: generates additional output
-m:	print name, Z, filter, observed mag and err for each star.

Required:

mm/dd/yyyy:	date for which constants are to be determined d may be a real number; y is full year, e.g., 1995.

The program prints error messages to stderr if trouble but usually tries to keep going. The resulting values are appended to the output file in the following format. No output is generated if there are not sufficient data to solve for all four sets of values.

JD (M/D/Y)

B: V_0 k' k''

V: V_0 k' k''

R: V_0 k' k''

I: V_0 k' k''

8.2.3 photom -- relative photometry of specified stars and asteroids

Photom is a program which performs aperture photometry on sparse-field star images. It is driven by a configuration file, described below. The config file lists several .fts image files and the locations of several fixed stars. The first image is a reference image and its last star is a reference star. The config file may also specify the location of a "wanderer" on two images and the location will be interpolated on all other images. All locations are given by their RA and Dec. The images must have WCS header fields.

Photom produces one line of output per image with the following columns:

- file number
- file name
- airmass
- pixel displacements from previous image
- value to be subtracted from JD to get heliocentric time
- Heliocentric JD (minus a bias of 2449000.0)
- Median sky value around reference star on this image
- RMS of sky value

Subsequent columns are in sets per star. Each set consists of:

- magnitude difference
- error estimate
- geometric mean of horizontal and vertical full-width-half-max values, pixels
- ratio of horizontal and vertical full-width-half-max ''FWHM' values, if -f
- X image location of brightest pixel in star, if -x
- Y image location of brightest pixel in star, if -x

In a given row, the magnitudes are relative to the star in the last column. The magnitude reported in the last column is relative that star on the first image. Estimates of the error of each brightness calculation are also given.

Usage:

Follows is a description of the command line options photom recognizes:

-c Crop: Find a bounding box that contains just the given stars and replace each image file with a version cropped to this size (plus a border of 20 pixels). The box is adjusted for the aligned location of the stars in each image so the location of the box will differ for each image but they will all have the same size box. Photom does this IN PLACE forever destroying the original image files. To record the cropping action that photom has performed it adds two integer fields to the FITS header of each such cropped image: CROPX and CROPY. The effective bounding box size is just the size of the image, of course.

-f Include the two FWHM columns; they are not printed by default.

-x Include the two X and Y columns; they are not printed by default.

-v Verbose: Generate extra output. One line per star per image consisting of:

star number

x= X location of star

y= Y location of star

max= maximum pixel value of star

r= radius where star ends and noise begins

sum= sum of all pixels within r

n= number of pixels within r

nmed= median pixel value in noise annulus

nmean= mean pixel value in the noise annulus

nsd= sd within the noise annulus

Follows is a description of the photom configuration file format:

All blank lines and lines which begin with # are ignored. Names which appear in BOLD FACE are keywords that should appear in the file; case is ignored.

The first line consists of two integer parameters, separated by one or more spaces:

rsearch: max radius to search for brightest pixel from given coordinate

apradius: the aperture radius to use, in pixels. If set to 0, the aperture will be automatically determined using the reference star on the reference image, based on that radius which maximizes the SNR. Use 0 if unsure.

The next section names the files to use:

FILES:

<file>

<file>

<file>

...

where:

<file> can be any of:

a simple file name;

a range; ranges must be of the form Xaa-bb.fts, where:

X is anything;

aa-bb is a starting and ending hex range, such as 10-a0.

or the name of a file containing file names or ranges. Must be

preceded with a * to mark it as a file of files.

There is no fixed limit to the number of files which may be given.

The next section defines the locations of stars and asteroids to measure:

FIXED:

<RA Dec>

<RA Dec>

<RA Dec>

...

where:

<RA Dec> are star coordinates. These may be in H:M:S and D:M:S format;

or decimal degrees (yes, even RA is given in decimal degrees).

WANDERER: file1 RA Dec file2 RA Dec

This line is optional. If present, it specifies the location of an object assumed to move linearly in time and space between its location on the two files indicated. The format of the RA and Dec fields are as for fixed objects, above. Its position on all other files will be interpolated from these known positions and times.

No more than 100 stars may be specified, including the wanderer if present.

Sample:

```
# Sample photom config file
10 10 .1
# the files
FILES:
gabb2114.fts
gabb2216.fts
gabb2218.fts
gabb2220-3f.fts

# the fixed stars
FIXED:
0:39:09.6 -0:26:37
0:39:45.2 -0:19:18
0:39:44.0 -0:32:00

# an asteroid
WANDERER: gabb2114.fts 0:39:22.7 -0:24.57 gabb2236.fts 0:39:15.0 -0:23:09
```

8.2.4 predict -- date and times of variable minima

Predict prints out a list of times when eclipsing binary stars will be at a minima. Only minima which occur during local night with the star above the horizon and within abs(HA) < 6 are shown, unless the -a switch is used.

The program requires two environment variables: LATITUDE, in rads +N, and LONGITUDE, rads +W.

Usage: predict [options]

-a:	show all minima, regardless of dusk/dawn/up/HA limits, etc.
-d m/d/y:	starting date (y is full year, as in 1996); default is today.
-f starfile:	alternate star list; default is ecl-bin.txt.
-n minima:	number of consecutive minima to report; default is 1.
-s starname:	name of star; default is all stars in star list file.

The program requires a file of known minima for the stars to be predicted. This file consists of 10 columns, as follows:

Name	the name of the star.
RA-h,m,s	three columns for RA hour, minute and second (J2000)
Dec-d,m,s	three columns for Dec degrees, minutes, seconds; the first of these may be immediately preceded with + or -
P	period, days.
JD0	Julian date of a minima, referenced to heliocentric time.
Mag	Magnitude range as Brightest-DimmestX, where X is a one-character band code such as B, V, R or I.

The output consists of several columns for each star and minimum in range, as follows.

Name	the name of the star.
Date	date of minimum, M/D/Y UT.
JD	time of minimum, JD.
Dusk	time of dusk the night of the minimum, UT.
UT	time of minimum, UT.
Dawn	time of dawn the night of the minimum, UT.
LST	time of minimum, Local Sidereal Time.
HA	Hour angle of star at minimum, H:M:S.
Elev	Elevation of star at minimum, Degrees.

8.2.5 snsearch -- supernovæ search

Snsearch "automatically" searches a list of test images against a list of corresponding reference images for stars that have appeared in the test image that are not in the reference image. The search region may be delimited by an area of interest.

Usage:

```
snsearch arcsec_sep config_file test1.fts ...
```

or

snsearch -f file_of_files arcsec_sep config_file

arcsec_sep is the maximum separation allowed between object locations, in arc seconds. It is also the amount by which the AOI is reduced for the test image.

config_file is the file containing the list of reference images. The file consists of one or more lines in the following format. A sample file, blink.idx, is located in $TELHOME/archive/config.

OBJECT ERa NDec WRa SDec reference.fts

...

where:

OBJECT: name of object, as it appears in the OBJECT FITS header field.

ERa: East RA limit in the form HH:MM:SS (J2000)

NDec: North Dec limit in the form DDD:MM:SS (J2000)

WRa: West RA limit in the form HH:MM:SS (J2000)

SDec: South Dec limit in the form DDD:MM:SS (J2000)

All lines which do not begin with an alphanumeric character are ignored.

The remaining arguments to snsearch are the test image file names. They are matched up with the reference images by matching OBJECT fields and so may be given in any order. A reference object must exist for each test object. The list of test files may also be in a file by using the -f option.

The output generated by snsearch is in two parts. The first part has one line per test image in the following format:

OBJECT rfile tfile date time nref ntest nsuper

where:

OBJECT: name of object, as it appears in the OBJECT FITS header field.

rfile: name of reference file

tfile: name of test file

date: date of test file (same as DATE-OBS FITS field)

time: time of test file (same as TIME-OBS FITS field)

nref: number of stars found inside AOI in the reference image

ntest: number of stars found inside AOI in the test image

nsuper: number of stars in test with no corresponding star in reference.

The first part is followed by a form-feed character (control-l).

The remainder of the file is a report of the location of each potential supernovae. There will be one line for each supernovae candidate in the following format:

OBJECT tfile RA Dec

where:

OBJECT:	name of object, as it appears in the OBJECT FITS header field.
tfile:	name of test file
RA:	RA of candidate (J2000)
Dec:	Dec of candidate (J2000)

A blank line is inserted after each set of objects.

8.2.6 vsmon -- variable star monitoring

Vsmon is a program for monitoring a set of variable stars over an extended period. The list of stars to monitor is specified in a catalog file. The format is detailed below. Vsmon scans a given directory of FITS images and checks each image against the catalog to produce one line of output for each image whose OBJECT field is found in the catalog. The format of the output is detailed below.

Usage:

usage: vsmon [options]

-b:	use bright-walk star search method. default is max-in-area.
-c file:	alternate catalog file. default is $TELHOME/archive/photcal/vsmon.lst
-i file:	alternate image directory. default is $TELHOME/user/images
-o file:	alternate output file; use - for stdout. default $TELHOME/archive/photcal/vsmon.out
-r rmax:	max radius to search for star, pixels. default is 12
-v:	verbose

The catalog file format consists of three lines per target star, as follows. There is one line for the target star, and one line each for the calibrator and check stars. All fields are separated by blanks. Blank lines and lines which begin with # are ignored.

C name RA Dec V B-V V-R V-I

K name RA Dec V B-V V-R V-I

T name RA Dec V B-V V-R V-I

where:

C/K/T:	The first column is one of the characters C, K or T. These serve to mark each entry as the Calibrator, Check, or Target, respectively. They must appear in the first column and always in the order C then K then T for each star group.

name:	The second column is the name of the star. Only the name of the Target star is used by vsmon for anything. No spaces are allowed.
RA:	The third column is the RA of the star. The format is H:M:S.
Dec:	The fourth column is the Dec of the star. The format is D:M:S.
V:	The last 4 columns are the magnitudes of the star in several ...
B-V:	bands. Column five is the magnitude in the V band; column six ...
V-R:	is the B-V magnitude; column seven is the V-R magnitude; and ...
V-I	column eighth is the V-I magnitude.

The output file consists of one line per file as follows:

FName OName UD UT HJD Z F Tag TErr TTrig KDiff KErr

where:

FName:	name of file
OName:	name of object
UD:	UT Date of observation, M/D/Y
UT:	UT Time of observation, H:M:S
HJD:	JD of observation, corrected to Heliocentric time
Z:	airmass
F:	filter code
TMag:	computed true magnitude of target star
Terr:	statistical error in TMag, in magnitudes
TTrig:	Y if TMag is >= the trigger threshold, else N.
KDiff:	measured - expected check star magnitude
Kerr:	statistical error in check star, in magnitudes

8.2.7 vssearch -- variable star search

Vssearch searches for variable stars in a set of FITS images as follows:

Given a set of images taken with a given filter scan each image and build lists of all qualifying stars found in each. Match the stars by position. Find the brightest star that is on the most number of images and use it as a calibrator star. Disregard all images which do not contain the calibrator. Find the magnitudes, V, and errors, E, of each star on each surviving image. Stars which have too much intrinsic error or are not on at least 3 images with the calibrator are discarded. For each of the N surviving stars, compute how well the magnitudes fit a straight horizontal line, allowing for their noise estimates, as follows:

Vbar denotes noise-weighted mean:

$$\bar{V} = \frac{\sum \frac{V_i}{E_i}}{\sum \frac{1}{E_i}}$$

D denotes average deviation from mean:

$$D = \frac{\sum \left| V_i - \bar{V} \right|}{N}$$

Q denotes figure of merit:

$$Q = \sqrt{\frac{\sum \left(\frac{V_i - \bar{V}}{E_i} \right)^2}{N}}$$

Stars with D below a threshold (set in config file, default is .1) are discarded. Remaining stars are sorted by increasing Q, i.e., increasing likelihood they are variable. Q is based on the mean squared error.

Usage:

vssearch [options] [files...]

Options:

-c file: override internal defaults with a config file (see below)

-f file: file of filenames

-r RA Dec RefMag MagLim: Specify an explicit calibrator star:

RA: H:M:S.S

Dec: D:M:S

RefMag: assigned absolute magnitude

MagLim: dimmest abs magnitude to report

-s: product spreadsheet output instead.

Note that filenames may be given as arguments or in a separate file, but not both.

The config file consists of the following nine parameters, one per line; blank lines and lines which begin with # are ignored.

1. max radius to search for brightest pixel, pixels, default is 4
2. photometric aperture, pixels, default is 5
3. max separation between images to be considered the same star, arc secs, default is 5
4. filter code, default is R (all other images will be silently ignored)
5. reject stars with more than this much noise error, default is .1
6. reject stars with more than this much mag estimate error, default is .1
7. reject stars with raw counts greater than this, default is 40000
8. reject stars whose average deviation from the mean is less than this, default is .1
9. reject stars which do not appear on at this many images, default is 3.

Output:

The default output of vssearch is a table with one row for each star that was found on at least 3 images along with the calibrator star. The calibrator star is always the first star shown. The columns are as follows:

I	ordinal
Q	figure of merit -- higher means more variable.
N	Number of images in which the star was found with the calibrator.
RA	RA of star (on first image encountered)
Dec	Dec of star (on first image encountered)

Then for each image, the following columns are printed:

V	The magnitude of the star with respect to the calibrator.
Verr	The error in the magnitude.

The format of the Spreadsheet output (-s) is just a huge table, one row per image. The first column is the Heliocentric Julian date of the observation - 2450000, then follows two columns per star: the magnitude relative to the calibrator and the estimated error in the magnitude. If the values are not available for any reason they are printed as 99.99 and 9.99, respectively. Stars that were not on at least three images with the calibrator are not listed at all.

8.3 System management

OCAAS includes several command line tools involved with system management, converting among various units and other miscellaneous tasks. These are useful directly from an *xterm* prompt, or from within custom scripts built for handling larger projects.

8.3.1 Units

degrad	<degrees minutes seconds> or <degrees:minutes:seconds> > to radians
raddeg	<radians> to degrees
radhr	<radians> to hours:minutes:seconds
hrrad	<hours minutes seconds> or <hours:minutes:seconds> to radians

8.3.2 Time

lst current local sidereal time, based on computer time. Requires either an environment variable LONGITUDE, in radians +W, or access to the telsched.cfg config file.

jd -t	print current UTC as YYYYMMDDhhmmss (OCAAS log timestamp format)
jd -T <JD>	convert the given JD to YYYYMMDDhhmmss (timestamp form)
jd	print current UTC as JD
jd MMDDhhmmYYYY.ss	convert UNIX date-input style format to JD
jd YYYYMMDDhhmmss	convert OCAAS log timestamp format to JD
jd *jd*	convert JD to UNIX UTC date-output style format: DOW MMM DD HH:MM:SS UTC YYYY

udp_time_client This is a program which contacts NIST for the current time and sets the computer software clock. It does *not* set the hardware clock; do that using /sbin/hwclock. You must be superuser to change either. This is useful for a remote observatory which can make at least occasional connections to the internet but which does not have a GPS receiver with which to maintain accurate local time.

8.3.3 Quick Stop/Start

killTel kills those OCAAS daemons and GUI programs which are directly related to operating the telescope and dome. These include xobs, shm, telrun, telescoped and camerad. If run with the additional argument *-all* it also kills camera, telsched and xephem.

startTel will start a basic set of OCAAS processes, including telescoped and xobs. If run with the additional argument *-all* also starts camerad, shm, camera, xephem and telsched.

telsecure can be used to quickly close down a remote site. It stops all batch processing and issues a command to $TELHOME/comm/Shutter.in to close the shutter.

8.3.4 Batch queue status

Telshow prints exactly the same information as *shm* in the same format, but displays it as lines of text one time to stdout and exits. This is handy for checking the status of a remote system accessible only by a slow-speed link. It also makes it trivial to build a simple cgi-bin status web page.

Usage:

telshow: [-sq]

-s	separate and label each basic section
-q	include info on observing queue

Telshowl is a variation of *telshow.* Its output and options are the same but it takes over the *xterm* from which it is run and shows a new status report updated once each second. It is really just telshow running under the control of the *screen* utility.

8.4 Data management

The following sections describe sample Perl scripts included with OCAAS to demonstrate how data can be managed from a robotic telescope system. Such systems generate a huge amount of data and thought to some scheme is absolutely necessary. These scripts can only serve as starting points and ideas; at best, they must be edited for your local situation.

8.4.1 Zip floppy

The two scripts *imread* and *imwrite* handle saving and retrieving data onto Zip drive removable media. This was developed to support a scenario where there was no real-time communication possible at all with the remote observatory, but there was a local technician who could perform routine tasks.

Each morning, the tech would run *imwrite.* It prompts for a Zip floppy. If the floppy contains a list of images and/or logs, imwrite removes them from the local hard disk. The floppy is erased. Then each image and log on the local hard disk are copied to the floppy. If the floppy fills, the operator is prompted to put in a new one. The above check for an existing list of images and logs is first performed, the floppy is erased, and the process continues until all images have been copied to Zip floppies. The tech is thanked and imwrite exits. The tech collects the Zips until a few have accumulated then mails them to the home institution.

At the home institution, the Zip floppies arrive every day or two. Personnel there run *imread.* It prompts for a Zip floppy. All images and logs on the floppy are copied to the local hard disk. As each file is transferred, its name is added to a list when it is confirmed to be safely on the hard disk. The floppy is erased and the list is copied on. The media is removed and sent back to the remote site to be used again.

8.4.2 *cdarchive* -- write a CD-R

The script *cdarchive* manages writing data to a CD-R (a writable CDROM). This was developed to support a scenario where there was only a low-bandwidth modem connection available to a remote observatory with a large format camera that generates hundreds of megabytes of data each night, but there was a technician available a few times a week who could load and mail media. It works in conjunction with *nogchkin* to remove data only after it has been confirmed delivered at the host site. This script will of course require some editing for use at your site.

This implementation of *cdarchive* expects five file systems, known as *staging areas,* to be at the paths /mnt/cdimage/d0 through d4, each just large enough to hold 650MB, the size of one CD-R. When first started, it scans each staging area for the fullest one which has not yet been burned. Files from several areas within $TELHOME are copied or moved into the selected area until either there are no more files or the area fills. If the area fills, it is marked with a BURNED stamp and is written to a CD-R which is assumed to be loaded with blank media. The writing process creates a temporary file of size 650MB at /mnt/cdimage/raw/image.raw.

The set of files which are moved out of the $TELHOME area to the same paths within the selected staging area are defined by the follow globs:

```
user/images/*.ft[sh]
user/logs/*.log
archive/images/*.ft[sh]
archive/userlogs/*.log
archive/pointmesh/*.ft[sh]
archive/logs/*.log
```

The following globs define those files which are only copied (and remain):

```
archive/config/*
archive/photcal/*
archive/calib/*.ft[sh]
```

A file is added to the selected staging area just before it is burned. The file has not contents and so can always be added even when the filesystem fills. The format of the file is as follows:

```
BURNED.YYYYMMDDHHMMSS
```

The actual burning process is performed by *mkisofs* version 1.12b3 and *cdrecord* version 1.6 both of which are included in the distribution of *xcdroast*, version 0.96d available on the web at

```
http://www.rz.fh-muenchen.de/htbin/htimage/home/ze/rz/services/projects/xcdroast/
d_xcdroast.conf?90,20
```

8.4.3 *cdrcv* -- read a CD-R written with cdarchive

This script, *cdrvc*, reads in a CD-R written by *cdarchive.* The first argument is required and is the serial number of the CD-R. Since this number becomes part of the permanent index, the script asks that it be confirmed. The script then asks whether to run *distemail*, as described below. The script mounts the CDROM drive if it is not already. Then it performs the following operations:

- copy and index newer archive/images to user/images
- copy newer archive/userlogs to user/logs
- copy newer archive/logs to archive/logs
- copy and index newer user/images to user/images
- copy newer user/logs to user/logs
- index (but do not copy) newer archive/calib
- index (but do not copy) newer archive/photcal
- index (but do not copy) newer archive/pointmesh
- copy BURNED* file to archive/logs

When these actions complete, the script umounts the CDROM if it mounted it. It may then run *distemail,* depending on how the query regarding same was answered when the script started.

Index refers to a text file, $TELHOME/archive/db/image_idx.list. This file contains the complete FITS header of each image. Each line of the header is prefixed with the CD-R serial number and the name of the image file (without a trailing .fts or .fth suffix). This file is expected to serve as the raw input to an SQL or other general purpose data base management system.

8.4.4 *distemail* -- distribute email about new images

The script *distemail* will send email to all users with new images informing them new image files are available. You will want to edit this script to suite your local situation. The script is based on the files in $TELHOME/user/images. The first three characters of the file name are considered to be the observer code. Codes, names and email addresses are obtained from the file *obs.txt* located in the ocaas login home directory. The format of the obs.txt files is four fields per line, separated by a pipe symbol (|). The four fields are:

- observer code
- email address
- name
- comments

The email field may contain more than one address separated by blanks but only the first will be used.

With no argument, distemail sends one mail message to each observer code with at least one file in $TELHOME/user/images. One optional argument is the maximum age of the file to consider as “new”, in hours. A record of each email sent is made in $TELHOME/user/logs/summary/emaillog.

8.4.5 *distimages* -- distribute images

This script, *distimages,* copies all image files and logs from $TELHOME/user to other places on the local network. It also removes all files and logs from the same area older than 7 days. You will want to edit this script for your local situation.

8.4.6 Modem control

Several sample scripts are included which help manage traffic to and from a remote site via modem using a ppp connection. They will of course need editing to fit your particular situation but they can serve as useful starting points and ideas.

nogchkin is designed to be run from a crontab script on a regular basis, say, hourly. Each time it checks in it performs the following tasks. In these descriptions “upload,” “download” and “local” are from the point of view of the telescope computer. All transfers are performed with ftp after logging in as user ocaas on the remote system. While performing each ftp transfer, the integrity of the ppp connection is monitored. If it fails, the connection is reestablished and the ftp is restarted. The link assumes the ISP will assign a dynamic IP on each connection. In order for the remote system to know this address and be able to log in to the telescope system, the file *iro-wmo-ip* is written in $TELHOME/comm and contains the IP. The file is removed when the script takes down the link.

- if the ppp connection is ok, exit

do {

- upload each image in user/images which has PRIORITY <= 0 then move to archive/images
- download archive/telrun/telrun.sls, if present
- if present, upload archive/telrun/telrun.sls as telrun.now
- upload each log in user/logs then move to archive/userlogs
- upload/append archive/logs/wx.log then append to local archive/logs/wx.all.log and remove wx.log
- for each archive/logs/BURNED* stamp file on the remote system delete all files in the local file system which contains the same stamp and delete the stamp on both sides. This feature works in conjunction with the *cdarchive* script to remove local data once it has been confirmed to arrive at the remote site.

} while (the file *nog_stayup* exists in $TELHOME/comm on the remote system)

nogppp is used by nogchkin to invoke pppd and establish a connection. An array of several different accounts can be set up to try in order if any account is busy or fails.

nog.chat is the chat script used by nogppp to perform the actual dialing operation.

pppoff shuts down the ppp connection.

8.5 System Tuning

These tools can be helpful during the installation phase of a new telescope.

8.5.1 mntmodel

This command-line tool reads a set of telescope axis calibration points and computes a best-fit model. The model and algorithm are the same as that described for *xobs* in its Calib Axes dialog but permits any number of points to be used. The format of the input required by this program is lines, each of which contains the following information separated by blanks or tabs:

```
Target HA, hours
Target Dec, degrees
Target HA - WCS HA, arcminutes
Target Dec - WCS Dec, arcminutes
```

It will be noted that this is identical to the default output format of the *pterrors* tool. The format of the model that is output from this program is identical to that which is required in the home.cfg file; that is, config file entries for HT, DT, XP, YC and NP.

The primary purpose of *mntmodel* is to compute a pointing model based on the set of images collected as a pointing mesh and compare it to the model computed by *xobs*. If they are significantly different, it means the additional data in the (supposedly denser) pointing mesh was not effectively captured by the stars which happened to be chosen during the calibration procedure using xobs. If after inspection the model from the set of mesh images is judged to be significantly different than the one taken with xobs, the output from *mntmodel* can be installed directly as follows:

```
pterrors *.fts | mntmodel >> $TELHOME/archive/config/home.cfg
```

This all assumes that the pointing mesh was taken without any prior mesh, that is, without a telescoped.mesh file in $TELHOME/archive/calib. If a new model is installed in this way, use *Reload* in *xobs* and take a new mesh.

Mntmodel may also use the same calibration points collected by *xobs* because *xobs* dumps the points to its log file, xobs.log, each time a new Solve function is performed. The format of this log dump is also exactly the form expected by *mntmodel.*

Usage:

mntmodel: [options]

Purpose: find best mount model from pointing data.

Synopsis: pterrors -t *.fts | mntmodel >> home.cfg

-H HT	set initial guess at HT (use after -d to override)
-D DT	set initial guess at DT (use after -d to override)
-X XP	set initial guess at XP (use after -d to override)
-Y YC	set initial guess at YC (use after -d to override)
-t t	set desired model tolerance, rads (use after -d to override)
-d	get all defaults from telescoped.cfg and home.cfg
-f	turn on GERMEQ_FLIP
-g	turn on GERMEQ
-v	verbose
-z	turn on ZENFLIP

8.5.2 Dynamical modeling

When initially configuring the telescoped.cfg file for a new installation, the values HMAXVEL, HMAXACC, HDAMP (and their Dec-axis counterparts) and POLL_PERIOD must be set. These values all interact. Together they establish the dynamical behavior of the telescope control loop. If they are set too sluggish, it may take unnecessarily long to acquire new targets and tracking may not be reliable near the zenith of Alt-Az mounts. If they are too lively, motors may sound awful, targets may be incorrectly acquired and tracking may not be accurate.

The *dynamics* tool can help select good initial values for this tuning process. It simulates a telescope trying to acquire and track a hypothetical target moving on a path of ever increasing acceleration. The target position is of the form y = sin(t**2). This trajectory gets progressively harder and harder to following and can show the conditions under which the telescope will lose track. *Dynamics* also models the effect of computer load on the telescope control loop by introducing a random amount of jitter in the sample time of 0 - 30 ms.

The goal of the simulation to is to find parameters which minimize the time to acquire the target and yet can maintain track under severe conditions.

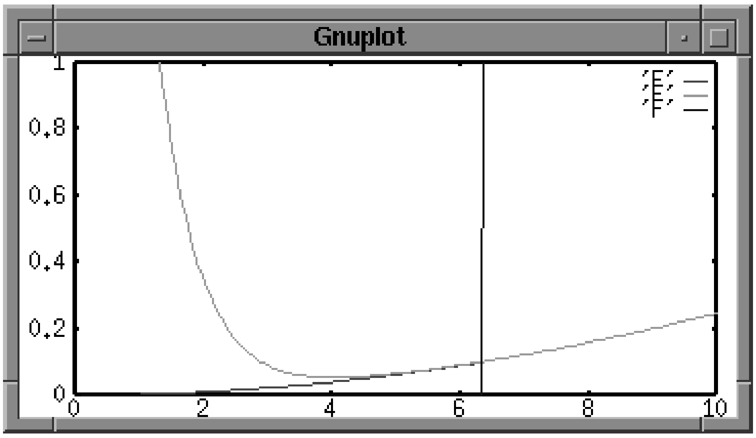
Start by setting MV to the maximum velocity at which the axis is to run under normal conditions. Set MA to the maximum acceleration the axis can tolerate under normal conditions. Set DT to 66 (corresponding to 15 Hz update rate). And set DF to .9 to start with. Run the program from an *xterm* and save the output to a temporary file, say, F, as follows:

```
dynamics 66 .25 .25 .9 > F
```

Now use *gnuplot* to see how well this performs. Start gnuplot and at its prompt type the following:

```
gnuplot> set data style lines
gnuplot> plot 'F' using 1:2, 'F' using 1:3, 'F' using 1:5
```

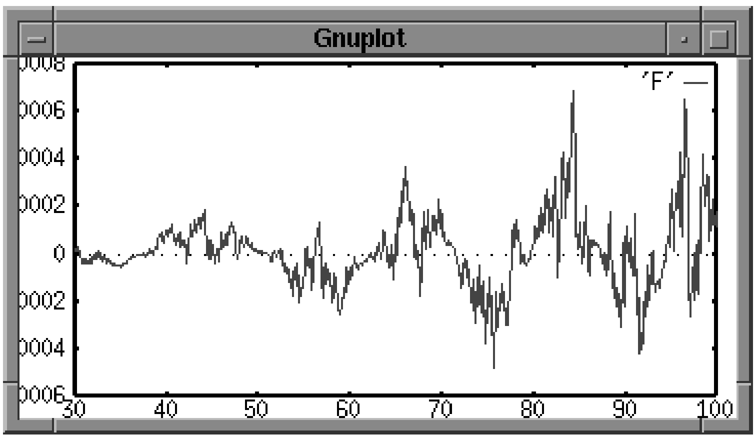
The graph is position error, in degrees, *vs.* time, in seconds. The red line is the hypothetical target. The green line is the telescope moving from a large distance away to acquire and lock onto the target. The blue line marks the first moment when the telescope matches the target within one arc second. Now use gnuplot's ability to zoom into a region of the graph as follows:




```
gnuplot> plot [0:10] [0:1] 'F' using 1:2, 'F' using 1:3, 'F' using 1:5
```

We see that tracking is accomplished in about 6 seconds when the scope is moving very rapidly from 5 degrees away.

Equally important is maintaining track. The hypothetical target is made to move in such a way that it is constantly increasing its acceleration. At some point, the telescope will not be able to keep up. The tracking error in degrees can be plotted as follows:




```
gnuplot>plot [30:] 'F' using 1:4
```

Check that the candidate configuration values can maintain track within a desired tolerance for as long as possible, keeping in mind that the accelerations near the end of the test scenario are very large in practice.

Run *dynamics* again with different values of DF and small changes in MA to find values which give small times to acquisition and smooth approaches without overshoot. But, make DF no smaller than necessary, that is, find the largest value at which making it smaller makes little difference in the time to acquire track.

Usage summary:

usage: DT SV SA DF

DT: sample interval, ms

MV: max velocity, rads/sec

MA: max acceleration, rads/sec/sec

DF: damping factor, 0..1

Output is 5 columns: T O S E L

T : elapsed time, 0..100 seconds

O : target position, degrees

S : scope position, degrees

E : error, O-S

L : 0 until E <= 1", then 5

8.5.3 logaxis

This is a simple tool which continuously reads the raw motor and encoder values for a given axis and prints the elapsed time and values. It is handy to investigate all sorts of mechanical issues with the mount electrical and mechanical components. It has two required arguments. One is the time between samples in ms. The second is the PC39 axis to monitor. It also has one optional argument. When set, it adds a fourth column to the output which is the value of the encoder as long as the scope is tracking, else 0. This is useful to investigate the dynamical hunting behavior when the system is acquiring any object by knowing the precise moment when the system believes it has tracking lock.

Usage: logaxis: [-l] <delay> <axis>

Purpose: log raw motor and encoder positions.

Optional arguments:

-l	add a fourth column to indicate whether scope is tracking

Required arguments:

delay	interval between polling, ms
axis	one char to indicate pc39 axis

Output format is T M E [l]

T	elapsed time since program start, seconds
M	raw motor counts, from RP command

E	raw encoder counts, from RE command
I	optional: E if tracking else 0

A handy adjunct to logaxis is the Perl script *autodiff.* This can be used, in conjunction with gnuplot, to plot the deviation of the motor/encoder values from a linear relationship.

To use these two tools, issue a constant velocity command to the axis motors then run *logaxis* for a while from an *xterm* as follows, and stop it with control-c:

```
logaxis 100 x > x.log
```

Then start *gnuplot* and type the following commands::

```
gnuplot>set data style linespoints
gnuplot>plot "<autodiff < x.log"
```

An perfect system will plot a straight horizontal line. Any vertical deviation indicates the amount by which the motor and encoder values are not tracking together. Both axes are in units of encoder steps.

8.5.4 PC39 commands

OCAAS uses the PC39 motion controller (or possibly other compatible models) from Oregon Micro Systems for all motor and encoder control. This section describes a few of the basic commands which can be issued directly to the PC39 for diagnostic purposes. To issue these commands, use an *xterm* and run the command *pc39.* It will issue a warning if OCAAS is also running. It is ok to run the pc39 program at the same time as OCAAS as long as you know what you are doing. If in doubt, use the *killTel* program to kill OCAAS first. When running along with OCAAS, each command must be preceded with the desired axis address; when running alone, this is not necessary since the PC39 will remember and continue to use the last axis addressed until changed. Though not always necessary, commands entered on one line are best separated by one space each.

Be especially careful using this level of interface. It is very low-level and you are working without a net.

In the table that follows, lower case letters should be typed exactly as shown; upper case letter denote values when are to be supplied as per the description.

Command	Mnemonic	Description
aX	address axis	address axis X
sa	stop axis	decellerate current axis to a stop
kl	kill	stop all axes instantly
jgN	jog	move at N steps/sec forever
mrN	move relative	move N steps at last jg rate then stop
maN	move absolute	move to motor step position N at last jg rate then stop

Command	Mnemonic	Description (cont.)
hm0	home	move in positive direction towards home position at last jg rate, reset counts to 0, stop
hr0	home reverse	like hm but moves in negative direction
hh	home high	home line is considered true when high
hl	home low	home line is considered true when low
he	home encoder	use encoder reference pulse as home, not an external switch
rp	report position	report current motor step count
re	report encoder	report current encoder count

9 Remote Access

If you have read the other chapters describing all the capabilities of OCAAS, it should now be clear how it fits together to allow fully automated operation of a remote observatory. OCAAS consists of nearly 200,000 lines of C written and continually improved over a period of four years. Each component is relatively simple. But putting the pieces together so the final collection is easy to use and reliable is the hardest part. We at CSI believe OCAAS accomplishes this feat handily.

Follows is a description of one scenario for setting up a remotely controlled system. The control scripts to do so are included in the OCAAS distribution. There are many choices to be made depending on the available bandwidth, degree of autonomous operation, available local assistance, roof control and so on.

9.1 One scenario

In this scenario, we assume the remote telescope communications is limited to a voice grade phone line. The remote telescope can dial an internet service provider (ISP) as a toll-free call. The remote site has control over a roll-off-roof and includes a GPS receiver and a weather station. The local operations center is on a live internet connection and operates a designated server at a fixed IP.

OCAAS can be set up so the remote computer operates as follows. When power is applied, the telescope is initialized to establish its home switch positions on all axes. The GPS receiver is activated to establish and maintain accurate time. The weather station data begin to be logged. The remote computer calls the ISP at the beginning of each hour to establish a PPP connection with a dynamically assigned IP from the ISP. Since the IP is not fixed, the remote computer performs an ftp operation to the designated server computer at the local center and writes a file containing its IP. In this way, the local center knows how to contact the remote computer if desired.

Each hour when the remote computer contacts the local server it checks for a new scan list file, telrun.sls in $TELHOME/archive/telrun, and downloads it. It uploads to the local server any image files with high priority and all engineering logs which have not yet been sent. It uploads the weather log accumulated since the last time of check-in. It also uploads the current state of the scan list file. Finally, it checks for the existence of a file in a prearranged location on the local server. If this file exists, the connection remains up until the file is removed. Otherwise the remote computer disconnects the call and repeats in one hour. Note that no long distance charges are incurred with this scenario.

Meanwhile, the remote telescope computer is working through the scan list file, telrun.sls, which it now has. It operates the telescope and camera to take the images, corrects them with the latest bias/thermal/flat fields, finds their WCS coordinates and FWHM statistics, and compresses them all according to options set in the scan list. It is also logging weather statistics to a file and watching for any parameters to pass thresholds. If a weather alert occurs, the roof is closed and further processing is suspended until the alert has not existed for a certain period then observing resumes. The roof is also opened at dusk or at the time of the first entry in the scan list for the current date, whichever is later; it is closed at dawn.

Each morning at a fixed time, a CD-R is written of the new images and logs. When the CD-R fills, it is ejected. An operator pops it out for mailing and puts in a fresh CD-R.

The effect of this is as follows. Personnel at the local center prepare a scan list from various and sundry *.sch* observing request files at their convenience using *telsched* and deposit it in the standard location. Soon the remote telescope checks in to find it and begins working on it. As it acquires new images, the high priority ones appear on the local server as if by magic, already fully corrected and calibrated, accompanied by their corresponding engineering logs. Over the course of a typical 12 hour night, the modem is capable of delivering some 100Mbytes of data. Weather data also appears regularly. Local users come in in the morning to find their data waiting. Web pages are able to display recent weather data in graphical form; provide users feedback as to the status of their requests, including the ability to retrieve their images via their web browser; and let surfers at large look at system activity. Web pages are also available for the creation and submission of individual observing requests. These are assigned a unique user code used to track images which it produces through the system.

The lower priority images also eventually make their way on to the local system via CD-R.

If the remote telescope is on a live internet connection, more data can be transferred of course. Plus, web pages on the local server can be built to inform users of current operating conditions, not just conditions in the recent past. The transfer of files can become more transparent by mounting the remote computers disks using NFS. *Xobs* and *camera* can be connected to a local X Server and used to operate the telescope in real time if desired. The X Windows protocol compression tool, *dxpc,* is included in the OCAAS distribution to aid in running these and other X programs over modem lines.

9.2 Screen shots

OCAAS is 100% capable of performing the above scenarios, and indeed is doing so as you read this. Follows are some screen shots of web interfaces developed by CSI for OCAAS customers. Many choices can be made and adjusted to fit your needs. Please contact CSI to discuss your remote operational requirements.

9.2.1 Schedule Submission form

Observer's code: [ ]

Project Title: [ ]

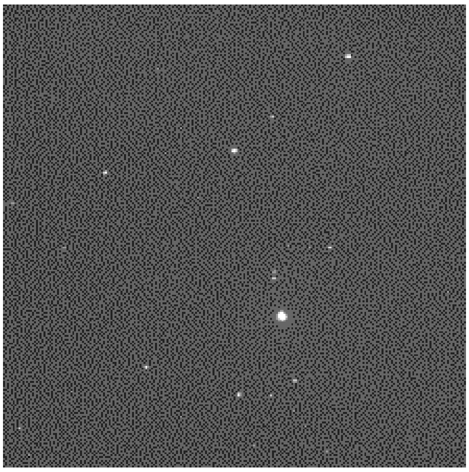
Observer(s): [ ]

Source List

Required Fields				Optional Fields		
Object Name	Catalog or Celestial Coordinates	Filter(s) and Duration(s)		Start Time	Repeat Count and Delay	Compression Scale and Additional Comments
[ ]	Use Catalog Position (Available Catalogs) Manual: 00:00:00 00:00:00 2000	V 60		[ ] LST HA	1 00:30:00	Compression: 0 Comments: [ ]
[ ]	Use Catalog Position (Avilable Catalogs) Manual: 00:00:00 00:00:00 2000	V 60		[ ] LST HA	1 00:30:00	Compression: 0 Comments: [ ]
[ ]	Use Catalog Position (Avilable Catalogs) Manual: 00:00:00 00:00:00 2000	V 60		[ ] LST HA	1 00:30:00	Compression: 0 Comments: [ ]
[ ]	Use Catalog Position (Avilable Catalogs) Manual: 00:00:00 00:00:00 2000	V 60		[ ] LST HA	1 00:30:00	Compression: 0 Comments: [ ]
[ ]	Use Catalog Position (Avilable Catalogs) Manual: 00:00:00 00:00:00 2000	V 60		[ ] LST HA	1 00:30:00	Compression: 0 Comments: [ ]

9.2.2 Last image acquired

Most recent image:



```
EXPTIME =                      30 / Exposur
OBJECT  = 'L002    '              / Object :
RA      = '0:55:05.4'             / J2000
DEC     = '0:50:01 '              / J2000
DATE-OBS= '27/11/97'              / UTC DD/:
TIME-OBS= '05:06:19'              / UTC HH::
```

Latest design update: 5 Sept 1996

Use your browser's "Go back" feature to go back.

9.2.3 System status

```
                     Clear Sky Institute

      Date :  MDY =  11/12/1998   JD = 2451129.81389
      Time :  LST =   4:50:48    UTC =   7:32:00
       Loc : NLat =  41:39:35  WLong =  91:31:48   El =  200.0m
   WX(Def) : Wind =  0KPH @   0  Air =  10.0C   0%RH 1010mB   0.0mm

   Telescope coordinates:
   Horizon :  Alt =  46:01:43     Az = 160:28:23   PA = -14:27:43
     J2000 :   RA =   5:44:32.8  Dec =  -0:36:56
       EOD :   RA =   5:44:29.8  Dec =  -0:37:00   HA =  -0:53:41.6
           :   RA =   5:44:21.0  Dec =  -0:33:55   HA =  -0:53:32.8
      Diff :  dRA =   0:00:08.8 dDec =  -0:03:05  dHA =  -0:00:08.8

   Offsets being applied:
      Mesh :  dRA =   0:00:00.0 dDec =   0:00:00  dHA =   0:00:00.0
   Jogging :  dRA =   0:00:00.0 dDec =   0:00:00  dHA =   0:00:00.0
     Sched :  dRA =   0:00:00.0 dDec =   0:00:00  dHA =   0:00:00.0

   Raw axis values:
     Az/HA :  Enc =  -0:02:42   Targ =  -0:04:40  Vel =  -0:00:21.0
           : NLim =-172:04:59   PLim = 179:19:41
   Alt/Dec :  Enc = -39:54:54   Targ = -39:51:22  Vel =   0:00:03.7
           : NLim =-150:45:11   PLim = 150:01:59
     Focus :  Mot =    -104.2   Targ =    -104.2  Vel =      0.0 um
           : NLim =    -131.3   PLim =      -6.6

   Telescope reference frame:
    T Pole :   HA =   0:00:00.0  Dec =  41:39:35 NonP =   0:00:00
    Ax Ref : XPol = 160:25:41   YHom =  85:56:37   R0 =   0:00:00

   Device status:
    'Scope : TRACKING
    Camera : Temp = -14C =   7F Targ = -20C  IDLE      Lights = 0
```

9.2.4 Total queue status

Queue state last updated Wed Nov 11 12:55:24 1998 UTC

Entries from 1998/11/11 02:00:00 through 1998/11/11 12:51:00 UTC

Obs Code	Obs Name	Title	Total Images	In Queue	Done	Failed	Percent Done
maa	Amanda Smith	3737period	2	0	2	0	100
mba	Kate Johnson, Heidi Arkebauer	andromeda	6	0	6	0	100
mbf	Gwen M Cassidy, David Owen	The Crab Nebula	4	0	4	0	100
mbn	Rob LaCroix	Rotation of Asteroids	25	0	24	1	96
mce	Owen Robertson, Reuben Merringer, John Denning	galaxies	5	0	5	0	100
mcf	Emily Mason, Erin Murphy	Properties of Asteroids	3	0	2	1	67
mcg	Lindsay Deere, Craig Dietrich	Var III Part 2	3	0	3	0	100
mci	Madeline Solien, Adrian Frey	M1	4	0	4	0	100
mcj	Katie Keith, Stephanie Hennessy	constellation	7	0	7	0	100
mck	Lindsay Deere, Craig Dietrich	Var III	12	0	12	0	100
med	Andrew Bessick, Korey Keninger	Binary Star #35	19	0	19	0	100
meg	Scott Enke	Asteroids	12	0	12	0	100

9.2.5 Individual queue status

Status for Observer code *apb* as of Thu Nov 12 07:19:25 UTC 1998 :

Schedules Submitted:

Schedules Accepted:

- apb307.sch
- apb308.sch
- apb309.sch

Images Queued:

Sched ID	File name	Object	UTC Date	Time	Filter	Exp

Images Completed:

***Image* key:**

- **S:** Display 128x128 gif image now
- **L:** Display 512x512 gif image now
- **F:** Retrieve as FITS image (extension should be .fts)
- **H:** Retrieve as HCompressed FITS image (extension should be .fth)
- **Transit:**Image is in transit from IRO.

Images No More Than 30 Days Old

Sched ID	File name	Image	Object	UTC Date	Time	Z	Filter	Exp
apb307	apb312f3	S L F H	star_X	1998/11/09	08:41:52	1.13	I	9
apb307	apb312f3	S L F H	star_X	1998/11/08	09:08:54	1.13	I	9
apb307	apb312f2	S L F H	star_X	1998/11/09	08:46:34	1.13	R	9
apb307	apb312f2	S L F H	star_X	1998/11/08	09:13:27	1.13	R	9

Index