Instrument Communication with the Magellan Telescopes

Glenn Eychaner – geychaner@lco.cl

Table of Contents

Starting communications

Communication with the telescope is established by opening a bidirectional TCP socket connection with the telescope server. The telescope server resides on host *mag1tcs.lco.cl* on Baade and *mag2tcs.lco.cl* on Clay; port `5800` is used for status requests only, and port `581x`, where `x` is the instrument rotator number, is used for status requests, wait requests, and commands.

It is recommended that the host names and port numbers be set in an initialization file or environment variable in the instrument software, rather than hard-coded into the code, so that they can be changed as necessary.

Authorization files

In order for an external computer to communicate with the telescope server, the canonical host name must be listed in the telescope server's list of allowed hosts for the port in question. The files containing these lists are located in `/usr/local/magellan/eds/mag1hosts` on *mag1tcs (Alpaca)* for Baade and `/usr/local/magellan/eds/mag2hosts` on *mag2tcs (Vicuna)* for Clay. These files are controlled and updated as needed by the Magellan programming staff, who will need to be informed of instrument machine name and IP address changes.

Command format

Commands, wait requests, and status requests are written to the TCP socket as strings terminated by a line feed ("`\n`") character. The telescope server likewise responds with a string terminated by a line feed. Commands and wait requests are differentiated from status requests by containing a space character (" ") in the string sent to the telescope server.

Table 1. Communications summary

Telescope	Server host	Status port	Command port	Authorization files
Baade	`mag1tcs.lco.cl`	5800	581x	`/usr/local/magellan/eds/mag1hosts`
Clay	`mag2tcs.lco.cl`	5800	581x	`/usr/local/magellan/eds/mag2hosts`

Status requests

The telescope server will respond to a status request with a string containing the requested value. The following tables contain the status requests that can be made, the format of the response, and a description of the status value. Status requests must **not** contain a space character in the string sent to the telescope server, or they will be misinterpreted as commands. In some cases, more than one different request will return the same status value; none of the request strings is preferred over the others.

Return Value Format

Returned values may contain leading or trailing spaces. Numeric values or number-like string values may be preceded by a sign character as required, with no spaces between the sign character and the first decimal digit.

Individual values

If the returned value is a decimal number, the number of significant decimal digits in the returned value is exactly as stated in the table, although leading and trailing zeros may be replaced by spaces or omitted. Otherwise, the value is a string, and is formatted exactly as shown in the table. If the requested value is not available (because the request is not understood, the system is rebooting, or the subsystem responsible for the value has failed or is shut down, for example), a terminated zero-length string is returned.

Multiple "dump" values

If the status request returns multiple values in a single string, the values will be separated from each other by one or more space characters (they are guaranteed not to contain space characters themselves). The individual values will be formatted as described in their descriptions, but no guarantees about field width in the returned string should be inferred; the string should be parsed by searching for the separating spaces rather than by slicing at exact character boundaries. Additional values may be added to the end of the string without notice as the needs of instruments and available data evolve, but the order of the existing values will not be changed without prior notice. If a requested value within the set of values is not known, it will be replaced by a series of asterisks (“ * ”).

Repeated Requests (Polling)

An instrument may request status values from the telescope as often as is necessary for the proper operation of the instrument, within certain guidelines described below.

Reusing sockets

When polling the telescope server for values, a client instrument should reuse the same socket rather than closing the socket and opening a new one for each value or polling loop. In addition, clients should be aware that it is impossible to request status values when a wait request is pending on the same socket; thus, a client will probably want to open a separate socket for wait requests, or check the `mountmv` and `gdrmountmv` status values instead of issuing a wait request. When moving the telescope, clients must use either a wait request or status values to check that it properly stops moving and report problems to the instrument user.

Update rates

Clients should be aware than no value in the telescope server is updated more frequently than 5 times per second, and many values are updated considerably less frequently; for example, the weather and environment values are currently updated only once per minute, and the catalog values are only updated when the telescope target is changed. If polling the telescope at high rates is desired, the document author can be contacted for specifics on how often each status value is updated.

Multi-value "dump" requests

Clients should use the "dump" requests which return multiple values whenever possible; they were added to the system specifically for this use case. If the instrument is requesting more than two values which can be found in a multiple-value request, the multiple-value request should be used and the return values parsed from it instead.

Table 2. Current telescope position information

Request	Format	Description
`dateobs`	`YYYY-MM-DD`	UT date in year, month, day format.
`ut, telut`	`HH:MM:SS`	UT time in hours, minutes, and seconds.
`st, telst`	`HH:MM:SS`	Sidereal time in hours, minutes, and seconds.
`ra, telra`	`HH:MM:SS.SS`	Right ascension in hours, minutes, and seconds.
`dec, teldc`	`DD:MM:SS.S`	Declination in degrees, minutes, and seconds.
`epoch, telep`	`YYYY.YY`	Equinox of current telescope coordinates.
`ha, telha`	`HH:MM:SS`	Hour angle in hours, minutes, and seconds.
`airmass, telam`	`A.AAA`	Observational airmass.
`telaz`	`AAA.AAAA`	Azimuth angle, in degrees.
`telel`	`EE.EEEE`	Elevation angle, in degrees
`zd`	`ZZ.ZZZZ`	Zenith angle, in degrees.
`telpa`	`PPP.PPPP`	Parallactic angle, in degrees.
`teldm`	`DDD`	Dome azimuth angle, in degrees.
`dmstat`	`DD`	Dome status (`0 = closed` ; `1 = open` ; `-1 = unknown`)
`telguide`	`ab`	`a` : `0` = not tracking, `1` = tracking; `b` : guider number of active guider, or `0` if not guiding
`gdrmountmv`	`abc`	Telescope and guider motion status (see below)
`mountmv`	`abcd`	Telescope and rotator motion status flags (see below)
`datetime`	`<string>`	Dump of `dateobs` , `telut` , and `telst`

Request	Format	Description
`telpos`	`<string>`	Dump of `telra`, `teldc`, `telep`, `telha`, `telam`, and `rotangle`
`teldata`	`<string>`	Dump of `roi`, `telguide`, `gdrmountmv`, `mountmv`, `telaz`, `telel`, `zd`, `telpa`, `teldm`, and `dmstat`

`gdrmountmv` telescope and guider motion status

`a` **(telescope mount status)**

0	mount not moving
1	mount moving
2	mount move rejected (not implemented)
3	mount limit reached (not implemented)
4-8	reserved for future use
9	mount hardware failure (not implemented)

`b` **and** `c` **(guider** `1` **and** `2` **status)**

0	guider not moving
1	guider moving
2	guider X-Y move rejected (due to limit or conflict)
3	guider X-Y limit encountered (not implemented)
4	guider Focus limit encountered
5-8	reserved for future use
9	guider hardware failure (not implemented)

`mountmv` telescope and rotator motion status (`0` = `false`, `1` = `true`)

a	telescope slewing
b	rotator in closed loop (servo) mode (normally `1`)
c	rotator in requested position

`d` rotator in requested position for required time

Table 3. Catalog postion inputs

Request	Format	Description
`catra`	`HH:MM:SS.SS`	Current catalog object right ascension.
`catdc`	`DD:MM:SS.S`	Current catalog object declination.
`catep`	`YYYY.YY`	Current catalog object equinox.
`catro`	`RRR.RRRR`	Current catalog object rotator offset angle, in degrees.
`catrm`	`TTT`	Current catalog object rotator offset mode; one of `OFF`, `EQU`, `GRV`, or `HRZ`.
`catobj`	`<string>`	Current catalog object name (up to 30 characters, containing no spaces).
`catdata`	`<string>`	Dump of all of the above data (6 items, in the above order)

Table 4. Future telescope position inputs

Request	Format	Description
`inpra`	`HH:MM:SS.SS`	Input coordinate right ascension.
`inpdc`	`DD:MM:SS.S`	Input coordinate declination.
`inpep`	`YYYY.YY`	Input coordinate equinox. Always the same as `telep`.
`inpha`	`HH:MM:SS`	Input coordinate hour angle.
`inpaz`	`AAA.AAAA`	Input coordinate azimuth.
`inpel`	`EE.EEEE`	Input coordinate elevation.
`inppa`	`PPP.PPPP`	Input coordinate parallactic angle, in degrees.
`inpdata`	`<string>`	Dump of all of the above data (7 items, in the above order) and `nrotoff`

Table 5. Secondary mirror (vane end) position

Request	Format	Description
`telfocus, telfc, vezset`	`FFFFFF`	Secondary mirror focus (Z axis) set (instrument) offset, in microns.
`vefocus, vezenc`	`FFFFFF`	Secondary mirror focus (Z axis) encoder reading, in microns.

Request	Format	Description
`vezima`	`FFFFFF`	Secondary mirror Z axis ima (Shack-Hartmann) offset, in microns.
`vezpsn`	`FFFFFF`	Secondary mirror Z axis psn (flexure) offset, in microns.
`vexset`	`FFFFFF`	Secondary mirror X axis set (instrument) offset, in microns.
`vexenc`	`FFFFFF`	Secondary mirror X axis encoder reading, in microns.
`vexima`	`FFFFFF`	Secondary mirror X axis ima (Shack-Hartmann) offset, in microns.
`vexpsn`	`FFFFFF`	Secondary mirror X axis psn (flexure) offset, in microns.
`veyset`	`FFFFFF`	Secondary mirror Y axis set (instrument) offset, in microns.
`veyenc`	`FFFFFF`	Secondary mirror Y axis encoder reading, in microns.
`veyima`	`FFFFFF`	Secondary mirror Y axis ima (Shack-Hartmann) offset, in microns.
`veypsn`	`FFFFFF`	Secondary mirror Y axis psn (flexure) offset, in microns.
`vehset`	`FFFFFF.FFF`	Secondary mirror H axis (rotation) set (instrument) offset, in arcseconds.
`vehenc`	`FFFFFF.FFF`	Secondary mirror H axis (rotation) encoder reading, in arcseconds.
`vehima`	`FFFFFF.FFF`	Secondary mirror H axis (rotation) ima (Shack-Hartmann) offset, in arcseconds.
`vehpsn`	`FFFFFF.FFF`	Secondary mirror H axis (rotation) psn (flexure) offset, in arcseconds.
`vevset`	`FFFFFF.FFF`	Secondary mirror V axis (rotation) set (instrument) offset, in arcseconds.
`vevenc`	`FFFFFF.FFF`	Secondary mirror V axis (rotation) encoder reading, in arcseconds.
`vevima`	`FFFFFF.FFF`	Secondary mirror V axis (rotation) ima (Shack-Hartmann) offset, in arcseconds.
`vevpsn`	`FFFFFF.FFF`	Secondary mirror V axis (rotation) psn (flexure) offset, in arcseconds.
`vedataenc`	`<string>`	Dump of the axis encoder readings (5 items, in the order `Z X Y H V`)
`vedataset`	`<string>`	Dump of the axis instrument offsets (5 items, in the order `Z X Y H V`)

Request	Format	Description
vedataima	<string>	Dump of the axis Shack-Hartmann offsets (5 items, in the order Z X Y H V)
vedatapsn	<string>	Dump of the axis flexure offsets (5 items, in the order Z X Y H V)
vedata	<string>	Dump of the instrument offsets and encoder readings (10 items; 2 per axis, in the order Z X Y H V)
vedata2	<string>	Dump of the Shack-Hartmann and flexure offsets (10 items; 2 per axis, in the order Z X Y H V)

Table 6. Rotator information

Request	Format	Description
telroi	R	Rotator of interest (rotators 0 to 5 are NASW, NASE, CASS, AUX1, AUX2, and AUX3 respectively).
rotangle, rotofh	RRR.RRRR	Current rotator offset angle, in degrees.
nrotoff	RRR.RRRR	Input coordinate encoder angle, in degrees.
rotatore	RRR.RRRR	Current rotator encoder angle, in degrees.

Table 7. Rotator graphics angles

Request	Format	Description
pangle	RRR.RRRR	Input coordinate rotator encoder angle, for current view mode
eangle	RRR.RRRR	Current rotator encoder angle, for current view mode
gangle	RRR.RRRR	Current rotator gravity angle, for current view mode
nangle	RRR.RRRR	Current parallactic angle, for current view mode
hangle	RRR.RRRR	Current rotator horizon angle, for current view mode
fangle	RRR.RRRR	Future rotator encoder angle, for current view mode
rotgraph	<string>	Dump of all of the above data (6 items, in the above order)

Table 8. Guider probe information

Request	Format	Description
guiderx*N*	XXXX.XXX	Probe *N* (1, 2, or 3) x position in millimeters.
guidery*N*	YYYY.YYY	Probe *N* (1, 2, or 3) y position in millimeters.

Request	Format	Description
guiderf*N*	FFFF.FFF	Probe *N* (1 or 2) focus position in millimeters.
p*N*filt	F	Probe *N* (1, 2, or 3) filter (0 = none, 1 = red, 2 = visual).
p*N*mask	M	Probe *N* (1, 2, or 3) mask (0 = none, 1 = Shack-Hartmann, 2 = coarse Shack-Hartmann [not used])
gdr*N*data	<string>	Dump of all of the above data (5 items, in the above order) for probe *N* (1 or 2)

Table 9. Guider camera information

Request	Format	Description
c*N*cur	XXXXYYYY	Camera *N* (1, 2, or 3) cursor 1 position, in tenths of an unbinned pixel.
c*N*xy*M*	XXXXYYYY	Camera *N* (1, 2, or 3) cursor *M* (2, 3, or 4) position, in tenths of an unbinned pixel.
c*N*box	XXXXYYYY	Camera *N* (1, 2, or 3) cursor 5 position, in tenths of an unbinned pixel.
ca*N*	DDD	Camera *N* (1, 2, or 3) angle in degrees.
fwhm*N*	FF.FF	Camera *N* (1, 2, or 3) FWHM value.
counts*N*	DDDDD	Camera *N* (1, 2, or 3) object counts value.
fwhm	FF.FF	30-second average FWHM value from the active guider .
counts	DDDDD	30-second average object counts value from the active guider.
seeing	<string>	Dump of fwhm, counts, and tair (from the telescope environment information below)

Table 10. Weather information

Request	Format	Description
wxtemp	TTT.TT	Outside temperature (degress Celcius).
wxpres	PPPP.PP	Outside pressure (millibars).
wxhumid	HHH.HH	Outside humidity (percent).
wxwind	VVV.VV	Outside wind intensity (mph).
wxwdir	DDD.DD	Outside wind direction (degrees).

Request	Format	Description
`wxdewpt`	`TTT.TT`	Outside dewpoint (degress Celcius).

Table 11. Telescope environment information

Request	Format	Description
`ttruss`	`TT.TTT`	Telescope truss temperature (degress Celcius).
`tcell`	`TT.TTT`	Primary mirror cell temperature (degress Celcius).
`tseccell`	`TT.TTT`	Secondary mirror cell temperature, skyward side (degress Celcius).
`tambient`	`TT.TTT`	Dome air temperature (degress Celcius).
`tair`	`TT.TTT`	Primary mirror air temperature (degress Celcius).
`telenv`	`<string>`	Dump of `wxtemp`, `wxpres`, `wxhumid`, `wxwind`, `wxwdir`, `ttruss`, `tcell`, `tseccell`, `tambient`, and `wxdewpt`

Commands

The telescope server will respond to a command with a string containing a zero (“ `0\n` ”) if the command was accepted, and a negative number (usually “ `-1\n` ”) if the command was rejected. Command arguments are separated from the command and each other with spaces. A command which takes no arguments must end with a space in order to be recognized as a command.

For convenience and robustness, commands listed in the tables below which take no arguments (except “ `h` ”) are not required to end with a space. Other commands listed in the complete subsystem documentation linked below which take no arguments **must** end with a space.

Command arguments

Unless otherwise noted, numeric command arguments are interpreted as single precision floating point values (and thus may contain a decimal point and fractional part) and may be negative, but exponential notation is not supported. (Internal calculations are carried out using double-precision floating point values to ensure sufficient precision. The actual telescope position, for example, is stored as an azimuth and elevation in double-precision radians.)

Telescope commands

The following tables contain the most commonly used commands; a complete list of commands available to the telescope control system is documented in the {base-manual-url}/tcs/tcs-usermanual.html[telescope control system manual].

Slewing or offsetting the telescope more than approximately **60 arcseconds (one arcminute)** without operator confirmation (by using the `opslew` command) is not permitted. Changing observing targets using this interface is not recommended; the operator should be performing this function using an {catalog-url}[observing catalog].

When moving the telescope, clients must either issue a wait request or check the `mountmv` and `gdrmountmv` status values to ensure that it properly stops moving, and report problems to the instrument user.

Table 12. Telescope position commands

Command	Arguments	Description
`ra` *H M S*	$0 \le H \le 23$ $0 \le M \le 59$ $0 \le S \le 60$	Input (without moving) new right ascension coordinate in hours, minutes, and seconds; hours and minutes are rounded to the nearest integer value.
`dc` *D M S*	$-89 \le D \le 89$ $0 \le M \le 59$ $0 \le S \le 60$	Input (without moving) new declination coordinate in degrees, minutes, and seconds; degrees and minutes are rounded to the nearest integer value.
`mp` *N*		Input (without moving) new coordinate equinox. Reported telescope positions will immediately shift to new equinox.
`az` *N*	$0 \le N < 360$	Input (without moving) new azimuth coordinate in degrees.
`el` *N*	$0 \le N \le 90$	Input (without moving) new elevation coordinate in degrees.
`nro` *N*		Input (without moving) new rotator offset angle in degrees.
`slew`		Slew telescope to input position.
`opslew`		Slew telescope to input position, with confirmation from the telescope operator.
`h`		Halt a slew in progress.
`ro` *N*		Immediately move rotator to new offset angle in degrees.
`rotref`		Set the rotator guiding reference point.

Table 13. Telescope offset commands

Command	Arguments	Description
`ofra` *N*		Input (without moving) right ascension offset in arcseconds.
`ofdc` *N*		Input (without moving) declination offset in arcseconds.
`ofep` *N*		Input (without moving) offset equinox (`0` = displayed equinox, `-1` = apparent equinox).

Command	Arguments	Description
`offp`		Move telescope to the input offsets.
`offm`		Move telescope to the negative of the input offsets.
`dr N`		Immediately offset rotator by input angle in degrees.
`aeg A E`	`-60 ≤ A,E ≤ 60`	Immediately offset telescope azimuth and elevation in arcseconds.

Table 14. Telescope focus commands

Command	Arguments	Description
`zset N`		Change telescope focus to a position at *N* microns.
`zstr N`		Offset telescope focus by *N* microns from current position.

Guider probe commands

Commands can be sent to the guider probe control system by preceding the guider probe command with the string " `gp` ". The following table contains the most commonly used commands; a complete list of commands available to the guider probe control system is documented in the {base-manual-url}/guider/guider-usermanual.html[guider probe control system manual].

Table 15. Guider probe commands

Command	Arguments	Description
`gp`*`COMMAND`*		Sends *`COMMAND`* to the Guider PC.
`gpxy`*`P N M`*		Move probe *`P`* (`1` or `2`) to coordinate (*`N`*, *`M`*) in the focal plane.
`gpxy`*`P`*`r N M`		Offset probe *`P`* (`1` or `2`) by amount (*`N`*, *`M`*) in the focal plane.
`gpf`*`P`*`a N`		Move probe *`P`* (`1` or `2`) to an absolute focus of *`N`* millimeters.
`gpf`*`P`*`r N`		Offset probe *`P`* (`1` or `2`) focus by *`N`* millimeters
`gprdr`*`P R D`*		Offset probe *`P`* (`1` or `2`, or `3` for both) in right ascension and declination in arcseconds.
`gpaer`*`P A E`*		Offset probe *`P`* (`1` or `2`, or `3` for both) in azimuth and elevation in arcseconds.
`gpfld`*`P`*		Set probe *`P`* (`1` or `2`) slides to full field.
`gpsha`*`P`*		Set probe *`P`* (`1` or `2`) slides to Shack-Hartmann lenslet array.

Guider camera commands

Commands can be sent to the guider camera control systems by preceding the guider camera command with the string "g*P*", where *P* is a, b, or c for camera *1*, *2*, or *3* respectively. The following table contains the most commonly used commands; a complete list of commands available to the guider camera control system is documented in the {base-manual-url}/gcam/gcam-usermanual.html[guider camera control system manual].

Table 16. Guider camera commands

Command	Arguments	Description
g*P*COMMAND		Sends *COMMAND* to guider camera *P*.
g*P*fone		Switch guider camera *P* to full-frame mode (guider off).
g*P*fthr		Switch guider camera *P* to subraster (guide) mode, calculate and move telescope.
g*P*xys *N X Y*	$1 \le N \le 5$ $0 \le X,Y \le 999.9$	Set cursor *N* to position (*X*, *Y*).
g*P*xyr *N X Y*	$1 \le N \le 5$ $\|X\|,\|Y\| \le 999.9$	Move cursor *N* by (*X*, *Y*) from current position.
g*P*mc *N*	$1 \le N \le 3$	Simulate a mouse click of button *N* (1 = left, 2 = right, 3 = both)
g*P*avf *N*	$0 \le N \le 99$	Set the leaky-memory averaging parameter for full-frame mode.
g*P*avg *N*	$0 \le N \le 99$	Set the leaky-memory averaging parameter for guide mode.
g*P*tf *N*		Set the exposure time in seconds for full-frame mode.
g*P*tg *N*		Set the exposure time in seconds for guide mode.
g*P*send		Send one frame over the ethernet link
g*P*send *N*		Send a frame every *N* seconds, or stop sending if *N* = 0.

Wait requests

The telescope server will respond to a wait request with a string containing a zero ("0\n") if the wait condition (e.g. telescope move) has completed, a one ("1\n") if the wait condition failed to complete before the specified timeout, and a negative number (usually "-1\n") if the wait request was rejected.

The following table contains a complete list of available wait requests. Wait times are specified in integer seconds.

Timeout values

The timeout specified along with a wait request is the *maximum* time the request is allowed to take to complete; if the wait condition is fulfilled before the timeout has expired, the wait request will return as soon as the condition is fulfilled. However, the frequency at which timeouts are checked and the network latency may cause the client to receive a timeout response slightly more than 0.1 seconds after the specified timeout value; therefore, the client should set any socket timeout values (to check for dropped connections) at least 0.2 second longer than the wait request timeout.

For historical reasons, requests with wait times of one or zero seconds do not wait at all, but send back an immediate response. A client which intends to use this feature to poll the telescope move status should instead use the `mountmv` or `gdrmountmv` status values, which return the same information.

Interaction between wait requests and other requests

It is impossible to request status values or send commands on a socket connected to the telescope server when a wait request is pending on the same socket; therefore, clients which are polling the telescope may want to open a separate socket for wait requests.

Table 17. Wait requests

Request	Description
`mountw` *N*	Wait up to *N* seconds for telescope move to complete. In general, the telescope can take up to 120 seconds to complete a move.
`mountew` *N*	Wait up to *N* seconds for telescope move to complete; returns the current `mountmv` status (.e.g. `0111` for a successful move)
`vanew` *N*	Wait up to *N* seconds for vane end (e.g. focus) move to complete.
`gdrw` *N*	Wait up to *N* seconds for guide probe move to complete.
`gdrew` *N*	Wait up to *N* seconds for guide probe move to complete; returns the current `gdrmountmv` status instead of the normal value.
`gdrmountw` *N*	Wait up to *N* seconds for guide probe and telescope move to complete.
`gdrmountew` *N*	Wait up to *N* seconds for guide probe and telescope move to complete; returns the current `gdrmountmv` status instead of the normal value.
`gcam`*P*`w` *N*	Wait up to *N* seconds for guider camera *P* telescope correction to complete.
`gstopw` *N*	Wait up to *N* seconds for guiding to stop, usually after sending a `g`*P*`fone` command.
`gdrlockw` *N*	Wait up to *N* seconds for guide star to be centered on guide camera.

Shack-Hartmann Images

Some instruments that provide their own guiders may want to send Shack-Hartmann images into the telescope primary mirror correction system.

In addition to sending Shack-Hartmann images, the instruments will also need to provide guider position information to the telescope control system. The interface for doing so will be documented in a future release of this document.

Starting communications

Communication with the Shack-Hartmann system is established by opening a unidirectional TCP socket connection with the Shack-Hartmann image receiver. The image receiver resides on host *mag1tcs.lco.cl* on Baade and *mag2tcs.lco.cl* on Clay; the receiver port is `570x`, where `x` is the guider number (between `1` and `3`) minus one. The image receiver can read images in either guider camera format or FITS format; we recommend using FITS format when possible.

Guider Camera Image format

Each Shack-Hartmann image is composed of a header block and a data block.

Header block

Each image begins with header block consisting of 10 long integers (32-bit values) in network byte order (see '`man byteorder`' on most UNIX systems for more information). Only the first four values are currently used (the rest are reserved for future needs):

Table 18. Header contents

Value	Contents
1	The X image dimension (nx) in pixels; must be less than 2048
2	The Y image dimension (ny) in pixels; must be less than 2048
3	A frame counter which increments from 101 to 999
4	The value 1 to indicate a Shack-Hartmann image
5–10	Reserved for future use; should contain 0

Data block

The header block is followed by the image data; `nx*ny` unsigned short integers (16-bit values), again in network byte order.

FITS Image format

The S-H system expects a standard signed 16 bit per pixel image with a `BZERO` of 32768.0 and a `BSCALE` of 1.0. In addition to the standard FITS header keywords, image senders may also send any of the following keywords, which correspond to the TCS information keywords above. Any keywords not present when the image is received will be read from the TCS.

Sample Magellan FITS header

```
SIMPLE  =                    T  /
BITPIX  =                   16  /
NAXIS   =                    2  /
NAXIS1  =                  500  /
NAXIS2  =                  500  /
BZERO   =              32768.0  /
BSCALE  =                  1.0  /
CAMERA  =                    2  / Guider number; determined by incoming port
FRAME   =                  347  / Frame number, 0000-9999
SH      =                    1  / 1 if image is a S-H image, otherwise 0
DATE-OBS=         '2012-04-10'  /
UT      =           '03:27:24'  /
ST      =           '11:59:43'  /
RA      =        '12:29:06.62'  /
DEC     =        '+02:03:15.5'  /
EPOCH   =              2000.00  /
HA      =          '-00:30:03'  /
AIRMASS =                1.176  /
ZD      =            31.813900  /
TELFOCUS=               000149  /
ROTANGLE=             190.5425  /
ROTATORN=                    4  /
ROTATOR =             -001.991  /
TELGUIDE=                   12  /
GUIDERX =              022.167  / Position of S-H probe
GUIDERY =             -057.852  /
MASK    =                    1  / Always 1
CA      =                  323  /
GUIDERX1=             -100.817  / Position of guiding probe
GUIDERY1=             -040.362  / Keyword contains guider probe number
END
```

Last updated 2020-06-28 15:26:55 -0400