MMTO mount control protocol.

Tom Trebisky

MMTO mount control protocol.
by Tom Trebisky

Published August 10, 2001

Revision History

Revision 0.2 10 Aug 2001
Initial Draft

Table of Contents

Chapter 1. Introduction

This document gives details of the network protocol used to control the motions of the 6.5 meter MMT telescope.

Motions of the telescope are controlled by a computer and associated hardware referred to as the "mount computer".

A number of different protocols control various aspects of the mount computer. This document describes only one of these, which provides all services which are exported to individuals outside the MMT organization. Additional protocols exist for purposes such as system maintentance and servo tuning, as well as experimental and developmental purposes, and these are not described here.

This protocol is implemented as a server running within the mount computer which makes itself available on TCP port 5240. All communication is in plain ascii text, so that it would be possible, although somewhat awkward to access the protocol via telnet.

Older versions of the protocol were not multithreaded and so required clients to make a connection, quickly send a command, and close the connection so that the single socket could be shared. The current server is multithreaded and has no such limitation. However, there is a limit on the number of connections that can be supported and a single client should make only a single connection to the mount. There is also a definite limit on total packet throughput, and any attempt to send commands to the server at rates significantly faster than one per second will cause serious trouble.

An introduction to the commands

There are 3 categories of commands: those that set parameters (informative), those that get parameters (inquisitive), and those that command actions (imperative).

All commands will get some kind of response from the server. Those that inquire about values will get a list of values separated by newlines (i.e. one value per line), ending with the string ".EOF" on the final line. Imperative commands, and those commands that set values will receive an acknowlegement as the string "OK" or perhaps "ERR".

Certain commands with potential of causing hazard to the telescope are marked as "reserved", and are not available even though they are documented as part of this protocol (this protocol is used internally, but is also offered to outside users as well).

Interogative commands return a list of values, one per line, with each line containing a tag word followed by the value associated with the tag. (If this is not clear, look at the examples below). The intention here is that applications can be written that do not depend on the order in which values are returned (although the order is in fact deterministic). More importantly, applications can be written that do not depend on how many values are returned.

Future changes to the protocol.

This protocol has a number of known deficiencies and plans exist to address these problems. These may cause software to break that is written to exactly conform to the specification given herein. If the following guidelines are followed by those who write software that utilizes this protocol, imcompatibilities are much less likely and in most cases will be smoothly accomodated.

After sending an interrogation string, the application should loop reading response lines until it sees the final ".EOF" tag. Lines containing tag,value pairs should be parsed, and the desired values extracted. Lines containing unknown tags should be ignored. Applications can be written that do not obey these guidelines, but they are far more likely to fail when protocol enhancements are made.

Command groups

- Status commands
- Motion commands
- Offset commands
- Maintenance commands
- Miscellaneous commands

It should be noted that the inquisitive commands return grouping of related parameters. Many parameters may be accessed via more than one command, and in every case, a parameter with the same tag references the same item. The only difference in value might be due to the parameter changing as a function of time.

Notice that all tags are unique. In other words, no context is required to determine what value is indicated by a tag. We could for example return a tag "vmax" within the "alt_servo" command, and then return the same tag "vmax" within the "az_servo" command. We do not do this, as this would require knowing the command context to understand the tag, rather we give unique tags like "vmaxaz".

Chapter 2. Status commands

telstat

Name

`telstat` — obtain telescope status

Synopsis

telstat (inquisitive)

Typical Output

```
strack 0
drives 0
dalt 0
daz 0
drot 0
dbldg 0
cbrake 1
mbrake 1
stpin 1 0
.EOF
```

Explanatory comments

This command returns a variety of boolean status values as follows:

strack tells whether sidereal tracking is on.

drives tells whether any drives are on.

dalt tells whether the elevation drive is on.

daz tells whether the azimuth drive is on.

drot tells whether the rotator drive is on.

dbldg tells whether the building drive is on.

cbrake tells whether the elevation cell brake is set.

mbrake tells whether the elevation motor brake is set.

stpin gives the elevation stow pin status in a curious way.

The first variable tells if the pin is in.

The second variable tells if the pin is out.

(It can be neither meaning it is in motion or faulty).

> **Warning**
>
> Having two values returned for stow pin status is bad, this will be changed in the next revision of the protocol.

telpos

Name

`telpos` — obtain telescope position

Synopsis

telpos (inquisitive)

Typical Output

```
az 0.000000
alt 0.000000
rot 0.000000
pa -0.095752
ha +86:31:23.65
lst 11:22:21.22
ra 86:31:23.65
dec +86:31:23.65
.EOF
```

Explanatory comments

This command returns a collection of telescope positional values as follows:

az current telescope azimuth (fractional degrees).

alt current telescope elevation (fractional degrees).

rot current rotator position (fractional degrees).

pa current rotator position angle (fractional degrees).

ha current hour angle (HMS).

lst current local sidereal time (HMS).

ra current telescope right ascension (HMS).

dec current telescope declination (DMS).

catpos

Name

`catpos` — obtain current catalog coordinates

Synopsis

catpos (inquisitive)

Typical Output

```
cat_id FK5-0002
cat_ra 0.15297083
cat_dec 59.14977221
cat_rapm 6.82700000
cat_decpm -18.09000000
cat_epoch 2000.00000000
```

```
cat_coord J
.EOF
```

Explanatory comments

This command returns the last catalog coordinates sent to the mount via the "star" command. RA is in fractional hours, DEC is in fractional degrees.

cat_id object name from catalog.

cat_ra current catalog right ascension (fractional hours).

cat_dec current catalog declination (fractional degrees).

cat_rapm current catalog RA proper motion (fractional).

cat_decpm current catalog DEC proper motion (fractional).

cat_epoch catalog epoch (fractional years).

cat_coord "J" or "B"

times

Name

`times` — obtain time information

Synopsis

times (inquisitive)

Typical Output

```
lst 11:36:23.76
ut WED AUG 08 14:50:04 2001
jd 2452130.40978
.EOF
```

Explanatory comments

This command returns current time in 3 forms

lst - local sidereal time.

ut - universal time.

jd - julian date.

> **Warning**
>
> Due to an as yet unsolved bug, ut returns the local time, not universal time,

compos

Name

`compos` — obtain commanded servo positions

Synopsis

compos (inquisitive)

Typical Output

```
comaz 0.000000
comalt 0.000000
comrot 0.000000
.EOF
```

Explanatory comments

This command returns commanded servo positions of the 3 telescope axes in fractional degrees.

alt_servo

Name

`alt_servo` — obtain elevation servo information

Synopsis

alt_servo (inquisitive)

Typical Output

```
curalt 0.000000
comalt 0.000000
objalt 89.835658
velalt 0.000000
vincalt 0.000000
vincalt2 0.000000
dalt 0
modealt tracking
tflagsalt G
sflagsalt
perralt 0
scmdalt 0
vmaxalt 0.000000
.EOF
```

Explanatory comments

This command returns values associated with the elevation axis:

curalt current position (fractional degrees).

comalt commanded position (fractional degrees).

objalt object elevation if tracking (fractional degrees).

velalt velocity from abs encoder (fractional degrees/second).

vincalt velocity from incr encoder (fractional degrees/second).

vincalt2 velocity from incr encoder (fractional degrees/second).

dalt elevation drive status

modealt motion mode (tracking or slewing or ...)

tflagsalt TCS servo flag string.

sflagsalt LM628 status flag string.

perralt position error (encoder count units).

scmdalt servo velocity command (scaled encoder units).

vmaxalt velocity limit (degrees/second).

az_servo

Name

`az_servo` — obtain azimuth servo information

Synopsis

az_servo (inquisitive)

Typical Output

```
curaz 0.000000
comaz 0.000000
objaz 184.646340
velaz 0.000000
vincaz 0.000000
daz 0
modeaz tracking
tflagsaz G
sflagsaz
perraz 0
scmdaz 0
vmaxaz 0.000000
.EOF
```

Explanatory comments

This command returns values associated with the azimuth axis:

curaz current position (fractional degrees).

comaz commanded position (fractional degrees).

objaz object azimuth if tracking (fractional degrees).

velaz velocity from abs encoder (fractional degrees/second).

vincaz velocity from incr encoder (fractional degrees/second).

daz azimuth drive status

modeaz motion mode (tracking or slewing or ...)

tflagsaz TCS servo flag string.

sflagsaz LM628 status flag string.

perraz position error (encoder count units).

scmdaz servo velocity command (scaled encoder units).

vmaxaz velocity limit (degrees/second).

rot_servo

Name

`rot_servo` — obtain rotator servo information

Synopsis

rot_servo (inquisitive)

Typical Output

```
currot 0.000000
comrot 0.000000
objrot -0.111635
velrot 0.000000
vincrot 0.000000
drot 0
moderot slewing
tflagsrot G
sflagsrot
perrrot 0
scmdrot 0
vmaxrot 0.000000
offroti 0.000000
offrots 0.000000
signrot 0
rawrot 0.000000
rot_pos1 0.000000
rot_pos1s 0
rot_pos2 0.000000
rot_pos2s 0
rot_limn -185.000000
rot_limp 185.000000
.EOF
```

Explanatory comments

This command returns values associated with the instrument rotator.

currot current position (fractional degrees).

comrot commanded position (fractional degrees).

objrot object angle if tracking (fractional degrees).

velrot velocity from abs encoder (fractional degrees/second).

vincrot velocity from incr encoder (fractional degrees/second).

drot azimuth drive status
moderot motion mode (tracking or slewing or ...)
tflagsrot TCS servo flag string.
sflagsrot LM628 status flag string.
perrrot position error (encoder count units).
scmdrot servo velocity command (scaled encoder units).
vmaxrot velocity limit (degrees/second).
offroti instrument offset (fractional degrees)
offrots sky offset (position angle) (fractional degrees).
signrot rotator sign (XXX - bogus)
rawrot raw rotator position from encoders.
rot_pos1 Tape 1 position (degrees).
rot_pos1s Tape 1 status.
rot_pos2 Tape 2 position (degrees).
rot_pos2s Tape 2 status.
rot_limn Negative (CCW) limit.
rot_limp Positive (CW) limit.

Chapter 3. Motion commands

setrotlimn

Name

`setrotlimn` — set CCW rotator limit

Synopsis

setrotlimn (informative, reserved)

Typical Use

```
setrotlimn
-45.0
```

Explanatory comments

Sets the CCW rotator limit. Value in degrees. returns OK, but:

ERR if the limit exceeds the ultimate CCW limit,

ERR if the limit exceeds the ultimate CW limit,

ERR if the limit exceeds the positive limit,

ERR if it would place the current rotator position outside the new range of limits.

setrotlimp

Name

`setrotlimp` — set CW rotator limit

Synopsis

setrotlimp (informative, reserved)

Typical Use

```
setrotlimp
45.0
```

Explanatory comments

Sets the CW rotator limit. Value in degrees. returns OK, but

ERR if the limit exceeds the ultimate CCW limit,

ERR if the limit exceeds the ultimate CW limit,

ERR if the limit exceeds the negative limit,

ERR if it would place the current rotator position outside the new range of limits.

star

Name

`star` — slew to new star position

Synopsis

star (informative/imperative)

Typical Use

```
star
FK5-1321
12:33:34.261
-12:49:48.77
-00.109
+005.25
2000.0
J
```

Explanatory comments

This command requests a telescope slew to a new object. The telescope will acquire and track the object. 7 lines of information follow the command as follows:

object name (up to 78 characters)

RA coordinate (radians)

Dec coordinate (radians)

RA proper motion

DEC proper motion

Epoch

"J" or "B" epoch designation.

returns OK and initiates motion to the object, or returns ERR if the object is inaccessible (below the horizon, or other motion limits).

notice that the catalog record (in standard MMT format) corresponding to the above star is:

```
FK5-1321        12:33:34.261 -12:49:48.77 -00.109 +005.25  5.58 G5  J2000.0
```

slew

Name

`slew` — move telescope

Synopsis

slew (informative/imperative, reserved)

Typical Use

```
slew
90.0
189.0
0.0
```

Explanatory comments

This command requests a telescope slew to a static telescope position. This is typically for engineering and maintenance use (such as stowing the telescope at the end of the night. Coordinates are in fractional degrees.

> **Warning**
>
> rotator motion may be removed from this command as we typically manipulate the rotator independently of telescope motions.

returns OK and initiates motion, or returns ERR if the coordinates requested and inaccessible.

slewrot

Name

`slewrot` — move rotator

Synopsis

slewrot (informative/imperative, reserved)

Typical Use

```
slewrot
22.0
```

Explanatory comments

This command requests a slew of the instrument rotator to a static position. This is typically for engineering and maintenance use. Coordinate in fractional degrees.

returns OK and initiates motion, or returns ERR if the position requested is inaccessible (beyond limits).

cancel

Name

`cancel` — stop telescope motion

Synopsis

cancel (imperative)

Typical Use

```
cancel
```

Explanatory comments

Stop telescope motion by commanding the servos to target the current position.

> **Warning**
>
> Note that the servos will decelerate and then return to the position at which the cancel was issued.

tracking

Name

`tracking` — start or stop sidereal tracking

Synopsis

tracking (imperative)

Typical Use

```
tracking
on
```

Explanatory comments

This command turns sidereal tracking on or off. Anything other than "off" or "no" is taken as "on".

Always returns OK.

trackingalt

Name

`trackingalt` — start/stop elevation axis tracking

Synopsis

trackingalt (imperative)

Typical Use

```
trackingalt
on
```

Explanatory comments

This command instructs the elevation servo to track the current object coordinate. When not in tracking mode, the axis is in slew mode and will target the last slew position. Anything other than "off" or "no" is taken as "on". Changing modes can produce large telescope motion!

Always returns OK.

trackingaz

Name

`trackingaz` — start/stop azimuth axis tracking

Synopsis

trackingaz (imperative)

Typical Use

```
trackingaz
on
```

Explanatory comments

This command instructs the azimuth servo to track the current object coordinate. When not in tracking mode, the axis is in slew mode and will target the last slew position. Anything other than "off" or "no" is taken as "on". Changing modes can produce large telescope motion!

Always returns OK.

trackingrot

Name

`trackingrot` — start/stop rotator axis tracking

Synopsis

trackingrot (imperative)

Typical Use

```
trackingrot
on
```

Explanatory comments

This command instructs the rotator servo to track the current object coordinate. When not in tracking mode, the axis is in slew mode and will target the last slew position. Anything other than "off" or "no" is taken as "on". Changing modes can produce large telescope motion!

Always returns OK.

drives

Name

`drives` — start or stop telescope drives

Synopsis

drives (imperative, deprecated)

Typical Use

```
drives
on
```

Explanatory comments

This command turns on (or off) the telescope drives for the elevation and azimuth axes. Anything other than "off" or "no" is taken as "on".

Always returns OK.

altdrive

Name

`altdrive` — stop or start elevation drive

Synopsis

altdrive (imperative)

Typical Use

```
altdrive
on
```

Explanatory comments

This command turns on (or off) the elevation drives. Anything other than "off" or "no" is taken as "on".

Always returns OK.

azdrive

Name

`azdrive` — stop or start azimuth drive

Synopsis

azdrive (imperative)

Typical Use

```
azdrive
on
```

Explanatory comments

This command turns on (or off) the azimuth drives. Anything other than "off" or "no" is taken as "on".

Always returns OK.

rotdrive

Name

`rotdrive` — stop or start rotator drive

Synopsis

rotdrive (imperative)

Typical Use

```
rotdrive
on
```

Explanatory comments

This command turns on (or off) the rotator drives. Anything other than "off" or "no" is taken as "on".

Always returns OK.

setrotoffi

Name

`setrotoffi` — set instrument offset for rotator

Synopsis

setrotoffi (informative)

Typical Use

```
setrotoffi
20.0
```

Explanatory comments

This command sets the value of the instrument offset for the rotator. (fractional degrees).

Always returns OK.

setrotoffs

Name

`setrotoffs` — set position angle for rotator

Synopsis

setrotoffs (informative)

Typical Use

```
setrotoffs
20.0
```

Explanatory comments

This command sets the value of the sky offset (position angle) for the rotator. (fractional degrees).

Always returns OK.

vmaxalt

Name

`vmaxalt` — set elevation axis velocity limit

Synopsis

vmaxalt (informative, reserved)

Typical Use

```
vmaxalt
1.2
```

Explanatory comments

This command sets the maximum slew velocity for the elevation axis. Value is in fractional degrees per second.

vmaxaz

Name

`vmaxaz` — set azimuth axis velocity limit

Synopsis

vmaxaz (informative, reserved)

Typical Use

```
vmaxaz
1.2
```

Explanatory comments

This command sets the maximum slew velocity for the azimuth axis. Value is in fractional degrees per second.

vmaxrot

Name

`vmaxrot` — set rotator axis velocity limit

Synopsis

vmaxrot (informative, reserved)

Typical Use

```
vmaxrot
1.2
```

Explanatory comments

This command sets the maximum slew velocity for the rotator. Value is in fractional degrees per second.

Chapter 4. Offset commands

coord_coor

Name

`coord_coor` — get overall offset values

Synopsis

coord_coor (inquisitive)

Typical Output

```
ra_propmo 0.00000000
dec_propmo 0.00000000
ra_presnuc 0.00000000
dec_presnuc 0.00000000
ra_total 0.00000000
dec_total 0.00000000
alt_total 0.00000000
az_total 0.00000000
.EOF
```

Explanatory comments

Offset values are all in arc-second units.

Chapter 5. Maintenance commands

panic

Name

`panic` — check mount status

Synopsis

panic (inquisitive)

Typical Output

```
panic OK
hwstatus OK
ikstatus 0000
.EOF
```

Explanatory comments

This command is used to obtain the overall status of the mount software. In the event of an error on startup, the mount will save a panic string which will be returned instead of "OK", and in general any attempt to utilize the mount when it returns such a status will give unpredictable results (or more likely no results).

echo

Name

`echo` — check communication

Synopsis

echo (imperative)

Typical Output

```
OK
```

Explanatory comments

This command should in all cases return "OK". It can used to verify that a network connection to the mount is in place and healthy.

version

Name

`version` — show software version

Synopsis

version (inquisitive)

Typical Output

```
New version
.EOF
```

Explanatory comments

This command will return any amount of information describing the currently running software version, perhaps including date of compilation, versions of various modules, and other such information.

info

Name

`info` — list running tasks

Synopsis

info (inquisitive)

Typical Output - circa 2001

```
  NAME        ENTRY       TID     PRI   STATUS      PC       SP      ER-
RNO  DELAY
  ---------- ------------ -------- --- ---------- -------- -------
- ------- -----
  tExcTask   _excTask       fdc2f4   0 PEND          7bca0   fdc25c       0      0
  tLogTask   _logTask       fd99f4   0 PEND          7bca0   fd9958       0      0
  tm_servo   _servo_task    f50b90  60 PEND          66884   f50b30  3d0002      0
  tm_prio    _prio_task     fa43ac  95 DELAY         4c5a8   fa4368   30065   4753
  tm_reboot  _reboot_task   fa221c 133 PEND          66884   fa20ac  3d0002      0
  t5232      _rem_task      f9c71c 136 PEND          66884   f9c52c  3d0002      0
  t5233      _rem_task      f9a64c 136 PEND          66884   f9a45c  3d0002      0
  t5234      _rem_task      f9857c 136 PEND          66884   f9838c  3d0002      0
  t5235      _rem_task      f964ac 136 PEND          66884   f962bc  3d0002      0
  t5236      _rem_task      f943dc 136 PEND          66884   f941ec  3d0002      0
  t5237      _rem_task      f9230c 136 PEND          66884   f9211c  3d0002      0
  t5238      _rem_task      f9011c 136 PEND          66884   f8ff2c  3d0002      0
  t5231      _rem_task      f9e7ec 137 READY         4cd40   f9dc5c  3d0002      0
  tm_net0    _net0_task     f8df2c 137 PEND          66884   f8ddb8  3d0002      0
  tShell     _shell         fa108c 171 PEND          66884   fa0d68  3d0002      0
  tRdbTask   _rdbTask       fb58f8 175 PEND          66884   fb57d8   d0003      0
  tRlogind   _rlogind       fba1cc 176 PEND          66884   fb9fd0       0      0
  tTelnetd   _telnetd       fb8328 176 PEND          66884   fb8258       0      0
  tNetTask   _netTask       fd5668 180 READY         4b4b4   fd5600       0      0
  tFtpdTask  _ftpdTask      fb3440 185 PEND          66884   fb3384       0      0
```

```
  tPortmapd  _portmapd      fb6e10 190 PEND          66884   fb6cdc       16     0
  trdated    _rdated        fa6c4c 250 PEND          66884   fa6b90        0     0
.EOF
```

Explanatory comments

This command returns the literal output of the VxWorks "i" command and shows all tasks running in the mount computer and a variety of information associated with them. It is very handy for diagnosing problems with the mount software, but is unlikely to be of any use to the general public. It does indeed include an initial blank line.

stack

Name

`stack` — show stack status of running tasks

Synopsis

stack (inquisitive)

Typical Output - circa 2001

```
  NAME        ENTRY        TID     SIZE   CUR  HIGH  MARGIN
  ------------ ------------ -------- ----- ----- ----- ------
  tExcTask     _excTask     fdc2f4    7988   152   236   7752
  tLogTask     _logTask     fd99f4    4988   156   224   4764
  tm_servo     _servo_task  f50b90    7856    96   656   7200
  tm_prio      _prio_task   fa43ac    7860    68   216   7644
  tm_reboot    _reboot_task fa221c    3760   368   904   2856
  t5232        _rem_task    f9c71c    7668   496  1032   6636
  t5233        _rem_task    f9a64c    7668   496  1032   6636
  t5234        _rem_task    f9857c    7668   496  1032   6636
  t5235        _rem_task    f964ac    7668   496  1040   6628
  t5236        _rem_task    f943dc    7668   496  1032   6636
  t5237        _rem_task    f9230c    7668   496  1032   6636
  t5238        _rem_task    f9011c    7668   496  1032   6636
  t5231        _rem_task    f9e7ec    7668  3240  4580   3088
  tm_net0      _net0_task   f8df2c    3764   372   908   2856
  tShell       _shell       fa108c    9000   804  4020   4980
  tRdbTask     _rdbTask     fb58f8    8664   288  1572   7092
  tRlogind     _rlogind     fba1cc    4988   508   716   4272
  tTelnetd     _telnetd     fb8328    4988   208   416   4572
  tNetTask     _netTask     fd5668    9988   104   916   9072
  tFtpdTask    _ftpdTask    fb3440   11988   188   396  11592
  tPortmapd    _portmapd    fb6e10    4988   308  1040   3948
  trdated      _rdated      fa6c4c    9992   188   396   9596
  INTERRUPT                           1000     0   264    736
.EOF
```

Explanatory comments

This command returns the literal output of the VxWorks "checkStack" command which shows the stack sizes and status of various mount tasks. Again, it is very handy for diagnosing problems with the mount software, but is unlikely to be of use to the general public.

mem

Name

`mem` — show memory usage of running tasks

Synopsis

mem (inquisitive)

Typical Output

```
FREE LIST:
  num     addr      size
  --- ---------- ----------
    1   0xffd8f8       3704
    2   0xfae2d8        108
    3   0xfafaec        532
    4   0xfa7af8       3016
    5   0xfa8f54      10460
    6   0xfa7620         16
    7   0xfa8900         88
    8   0xf8b4d0       4812
    9   0xf8965c       7252
   10   0xfafa60         20
   11   0xfa75dc         24
   12   0xfae224         20
   13   0xf737a0       1420
   14    0x95c70   15437592
   15   0xffc000       6376
   16   0xff8000       8184
   17   0xff4000       8184
   18   0xff0000       8184
   19   0xfec000       8184
   20   0xfe8000       8184
   21   0xfe4000       8136
   22   0xfe0000       8184


SUMMARY:
 status   bytes    blocks   avg block  max block
 ------ --------- -------- ---------- ----------
current
   free  15532680       22     706030  15437592
  alloc    631024     2995        210         -
cumulative
  alloc   1055724     3546        297         -
.EOF
```

Explanatory comments

This command returns the literal output of the VxWorks "memShow" command.

Again, it is very handy for diagnosing problems with the mount software, but is unlikely to be of use to the general public. It does indeed include an initial blank line.

time

Name

`time` — obtain boot time and current time

Synopsis

time (inquisitive)

Typical Output

```
WED AUG 08 12:33:40 2001
WED AUG 08 14:26:02 2001
.EOF
```

Explanatory comments

This command displays the local time when the mount computer was booted followed by the current local time.

Chapter 6. Miscellaneous commands

This section collects together a number of commands used for telescope maintenance, almost certainly not of interest to agencies outside of MMTO.

set_fdc

Name

`set_fdc` — set starting position for pointing run

Synopsis

set_fdc (informative)

Typical Use

```
set_fdc
45.0
120.0
```

Explanatory comments

Establish the starting coordinates for a subsequent pointing data raster scan. Values given are elevation followed by azimuth in fractional degrees.

returns OK

get_fdc

Name

`get_fdc` — get starting position for pointing run

Synopsis

get_fdc (inquisitive)

Typical Output

```
20.000000
10.000000
.EOF
```

Explanatory comments

Return the starting coordinates for a subsequent pointing data raster scan. Values given are elevation followed by azimuth in fractional degrees.

go_fdc

Name

`go_fdc` — go to first (or next) star in pointing run

Synopsis

go_fdc (imperative, reserved)

Typical Use

```
go_fdc
```

Explanatory comments

Go to the next star in a pointing data collection raster scan. This and the following commands are part of a facility for the collection of pointing data (fdc stands for "flexure data collection"), and are not intended for public use. This does cause telescope motion to a new star!

returns OK

new_fdc

Name

`new_fdc` — set starting coords for pointing run

Synopsis

new_fdc (informative, imperative, reserved)

Typical Use

```
new_fdc
45.0
120.0
```

Explanatory comments

Establish the starting coordinates for a subsequent pointing data raster scan, and go to the first star. Values given are elevation followed by azimuth in fractional degrees.

This does cause telescope motion to a new star!

returns OK

save_fdc

Name

`save_fdc` — save record of pointing data

Synopsis

save_fdc (imperative, reserved, deprecated)

Typical Use

```
save_fdc
```

Explanatory comments

Save the current position information as a record in the current pointing data file.

returns OK

pointing

Name

`pointing` — get record of pointing data

Synopsis

pointing (inquisitive)

Typical Output

```
CAT: FK5-0520        14:06:40.951 -36:22:12.03 -04.293 -051.90  2.06 K0  J2000.0
RAA:   21.9409652  178.3275447
AA:   21.9552623  178.6967598
TAA:   21.9583174  178.6931672
LST: 13:59:01.80
CRD: 14:04:57.04 -36:21:33.05
SVEL: 0.0003 0.0041
SERR: 11 8
.EOF
```

Explanatory comments

This command returns the lines which should be deposited in a pointing data file. This command replaces save_fdc. The format of this record is not a topic for this document. This command will probably return nothing at all if a pointing run is not in progress, but can be used at any time.

calrot

Name

`calrot` — calibrate rotator

Synopsis

calrot (imperative, obsolete)

Typical Use

```
calrot
```

Explanatory comments

returns OK, obsolete and dangerous.

rotzero

Name

`rotzero` — zero the rotator

Synopsis

rotzero (imperative, obsolete)

Typical Output

```
rotzero
```

Explanatory comments

returns OK, obsolete and really dangerous.

setrotdir

Name

`setrotdir` — set rotator direction

Synopsis

setrotdir (informative, obsolete)

Typical Use

```
setrotdir
```

Explanatory comments

returns OK, obsolete, and exceptionally dangerous.

coord_corr

Name

`coord_corr` — debug coordinate conversion

Synopsis

coord_corr (inquisitive, reserved)

Typical Output

```
...
```

Explanatory comments

Returns various undocumented aspects of coordinate conversion (for maintenance and internal use).

Appendix A. Concrete Example of use of the protocol:

The following script accesses the protocol on port 5240 and displays the strings returned on standard output. It was used to obtain most of the example output used in the command descriptions given previously.

For example "showme telpos" would show the response lines (up to but not including the ".EOF") for the telpos command.

```
#!/usr/bin/perl -w
# $Id: remote.sgml,v 1.2 2002/05/03 05:30:26 tom Exp $

use IO::Socket;
use strict;

# 3/22/2001 incorporate switches from "im"

# defaults ...

my $test = 0;
my $verbose = 0;
my $thing = "info";

my $host = "mount";
my $port = 5240;

#my $host = "kofa";
#my $port = 13; # daytime

while ( $_ = $ARGV[0] ) {
    last unless /^-/;
    shift;
#   print "Arg: $_\n";

    $test = 1 if /^-t/;
    $verbose = 1 if /^-v/;

    $host = "mmtcell" if /^-c/;

    $thing = "stack" if /^-s/;
    $thing = "time" if /^-t/;
    $thing = "mem" if /^-x/;

    $thing = "telpos" if /^-p/;
    $thing = "telstat" if /^-m/;
    $thing = "cellstat" if /^-n/;
}

# override host or port with environment variable.

if ( $ENV{MOUNT_HOST} ) {
    $host = $ENV{MOUNT_HOST};
}

if ( $ENV{MOUNT_PORT} ) {
    $port = $ENV{MOUNT_PORT};
}

# final override from command line
if ( $test ) {
    $host = "mmvme2";
}

if ( defined $_ ) {
    $thing = $_;
}

print "Asking $host (on port $port) for $thing\n" if $verbose;

my $sock = IO::Socket::INET->new (
  PeerAddr => $host,
```

```
  PeerPort => $port,
  Proto => 'tcp'
     );

die "No socket!! (Reason: $!)\n" unless $sock;

print $sock "$thing\n";

while ( defined($_ = <$sock>) ) {
    last if /^.EOF/;
    print;
}

close ($sock);

# THE END
```