TCS3 Command Reference

1. Introduction

This document describes the set of commands available to the user through the TCS3 telescope control software. These commands may be entered directly on the Command Line Interface (CLI) of the TCS3 MCC graphical user interface screens, through a telnet connection, or indirectly through the use of widgets on the MCC screens or the t3_remote interface.

2. Summay of Commands by Catagory

2.1 Observing and Servo

Add.Offset.opt – Options when adding offset to base position.
Autopid – Enables or disable the autopid function.
Autopid.set – Specifies a set of PID coefficients.
Base - Load a new position into the position table base.
Base.inc - Increment the RA, Dec values of the base.
Beam.set - Specify the RA and Dec offsets for the beam
Beam.init – Initialize the beam offset.
Beam.inc - Apply an increment to the beam RA and Dec offsets
Beam.on or **Beam.B** - Enable application of the beam RA and Dec offsets
Beam.off or **Beam.A** - Disable application of the beam RA and Dec offsets
Beam.toggle - Toggle the beam offset enable status
Cat.search - Search catalog for the star closest to given RA/Dec within a specified radius
Cat.index - Search specified catalog by index and load results into 'next' buffer
CS - Set the default coordinate system
Epoch - Set the default epoch
Equinox - Set the default equinox
HSlew.Limit.Override – Override horizontal slew velocity limits.
Info - Return a selected subset of tcs information
Next.Clear - Clear a next star entry
Next – enter object data into the 'next' buffer (using RA, Dec coordinates).
Next.hadec – enter object data into the ‘next’ buffer using HA Dec coordindates.
NS.rate - Specifies an non-sidereal rate to be applied to the base position.
NS.rate.inc - Increments the base non-sidereal rate.
OS.2base - Transfer the enabled offsets and rates to the base and clear the offset values
MP - Execute a Motor Position move
MP.cnt - Specifiy the MP destination position in motor counts
MP.inc – Increments the MP destination position in arcseconds.
MP.Vel - Set the velocity for a motor position move
PID.Dec – Makes a requests to set the PID for the Dec axis.
PID.HA – Make a request to set the PID for the HA axis.
Polar.Motion - Specifies the earth's polar motion

MV - Execute a Motor Velocity move
Mw.zero – zero the measurement widget value.
Scan.Clear - Remove scan offsets to return to the original position
Scan.Go – Scans by moving to the offset ra, dec position.
Scan.Return – Scans by returning to the 0,0 offset position.
Scan.Set - Set up the parameters for scanning to an offset position
Slew - Slew to the next object in the specified buffer
Slew.Abort - Abort a slew and switch to track mode
Slew.Reslew – Re-issue a the slew command to the pmac (to fix "slew not slewing" problem).
Stop - Put the tcs system in the 'stop' mode
Sw.Limits.Override – Overrides software limt velocity controls.
Sw.Limits.Set – Set the software limits.
Track - Commands tcs to enter the 'track' servo mode
User.Inc - Apply an increment to user RA and Dec offsets
User.Init – Initialize the User Offsets.
User.Off - Disable user offsets
User.On - Enable user offsets
User.Set - Set the values for the user RA and Dec offsets
User.Toggle - Toggle the user offset enable setting
User.Spiral.Center – Move to the center of the user spiral..
User.Spiral.in – Move inwards on the user spiral trajectory.
User.Spiral.out – Move outwards on the user spiral trajectory.
User.Spiral.rate – Specify the velocity of the user spiral.
User.Spiral.stop – Stop moving on the user spiral.
User.Spiral.wid – Specify the with of 1 rotation of the user spiral
UT1Delta - Set the value of ut1 - utc.
Wavelength- Set the value of the observed wavelength

2.2. Facility Hardware Control

Ape.mode – Set the fio_ape execute mode.
Ape.Pos - Manual method for initialing the APE value.
Ape.set.pmac - Tells the tcs to initialize the position in the PMAC servo controller using the APEs..
Collimate.EW.Dpos- Set desired East/West collimation position
Collimate.EW.Dpos.Inc- Increment East/West collimation position
Collimate.NS.Dpos - Set desired North/South collimation position
Collimate.NS.Dpos.Inc – Increment North/South collimation position
Collimate.enable - Enable or disable collimation control loop
Collimate.table.read – Loads collimation table data into tcs3.
Collimation.table.set – Set the collimation desired position using the collimation table.
CW.Dir - Set the directory path for counterweight files
CW.Dpos- Set selected counterweight to position described as a voltage
CW.Enable - Enable or disable counterweight movements
CW.Manual.cntl - The CW command to set a counterweight up, down or off
CW.Mode - Select the counterweight operational mode
DHP.**enable** – Enables the DOME handpaddle.

DHP.**swapEW** – Swaps East and West inputs for the Dome HP.
DHP.**swapNS** – Swaps North and South inputs for the Dome HP.
DHP.**rate** – Sets the velocity for the Dome HP in as/s.
Dome.Auto.Offset – Set the an offset angle during auto (tracking) mode for dome control.
Dome.HP.Speed – Sets maximum speed for dome handpaddle mode.
Dome.Manual -Control dome motion while in manual mode
Dome.Manual.Speed – Sets the maximum speed for dome manual mode.
Dome.Mode - Set the dome movement control mode
Dome.Speed - Set the dome movement speed factor
Focus.Adj.Enable – Enables or disable the Temperatue and Position adjustment to the focus.
Focus.Enable - Enable or disable the focus control loop
Focus.Dpos - Set the desired focus position
Focus.Dpos.Inc - Increment or decrement the focus position
Hexapod.init – runs the Hexapod initialize maco file.
Humidity.Wn - Set relative humidity level to trigger warning.
Mirror.Cover - Open or close the mirror cover
MS.Support – Turn mirror support off or on.
MS.Fault.Ack - Re-enable mirror support after a fault
MC.Actuator – Mirror cooling fan control.
MC.Fan – Mirror cooling fan control.
MC.Heater – Mirror cooling Heater control.
MC.Mode – Sets the Mirror cooling mode.
MC.SetPt – Sets Mirror cooling set point temperature.
OH1.**enable** – Enables the OH1 handpaddle.
OH1.**swapEW** – Swaps East and West inputs for the OH1.
OH1.**swapNS** – Swaps North and South inputs for the OH1.
OH1.**rate** – Sets the velocity for the OH1 in as/s.
OH2.**enable** – Enables the OH2 handpaddle.
OH2.**swapEW** – Swaps East and West inputs for the OH2.
OH2.**swapNS** – Swaps North and South inputs for the OH2.
OH2.**rate** – Sets the velocity for the OH2 in as/s.
SafetyBrd.Reset – Reset the error latch on the safety board.
Secondary – Identifies the secondary in use (chopper or hexapod).
Shutter.**Lower** - Control the lower shutter
Shutter.Upper - Control the upper shutter
System.power – Turns system power on or off.
Track.NoIOnOffset.HA – Controls the integrator hold option during tracking.
Track.NoIOnOffset.Dec – Controls the integrator hold option during tracking.
TOHP.enable – Enables the TO Handpaddle inputs.
TOHP.rate – Sets the velocity form the TOHP in "/s.

2.3. Environmental and Other setup

Elevation - Set the elevation of the observatory
Humidity - Manually set the relative humidity
Latitude - Set the latitude of the observatory

LTOffset - Set the ut to local time offset
Pressure - Specifies the atmospheric pressure
TempK - Set ambient temperature value in degrees Kelvin
VTCS.Env.Update – Enables enviroment data updates from FIO inputs.

2.4. Laser Traffic Control

Ltc.Enable - Enable/disable Laser Traffic Control updates
Ltc.Filename - Set the full path for the LTC file name
Ltc.Fov - Set Laser Traffic Control Field of View
Ltc.Impact - Set the value for the Laser Traffic Control 'impact' field
Ltc.Period - Set period between LTC file updates

2.5. Pointing Map

Pt.Add.Star - Append the pointing data for the current star data to the tpoint file
Pt.Caption - Specify the text of a caption for the pointing data file
Pt.Clear – Clears all user pointing offsets and rates.
Pt.Convert - Converts current HA & Dec to numbers to enter into t3remote to test pt runs
Pt.Dir - Define the directory path for pointing data files
Pt.Find - Find the nearest guide star to the given HA and Dec and load into the next object buffer.
Pt.MAdj – Add pointing corrections (peak, spiral, rates) to the map via the MAdj registers.
Pt.MAdj.Set – Set the IH/ID values of the madj pointing register.
Pt.Map - Turn the pointing map on or off
Pt.Map.Set - Set a pointing map coefficient value
Pt.Next - Get the next (or specified) pointing object HA and Dec from table
Pt.Open - Open the pointing data file (see pt.dir and pt.filename)
Pt.Peak.Clear - Clear the Peak IH/ID pointing coefficient values.
Pt.Peak.Set - Set the Peak IH/ID pointing coefficient values.
Pt.Peak.Inc - Increment the value of the Peak IH/ID coefficients
Pt.Restore – Restores the last saved MAdj IH, ID values saved by pt.save.
Pt.Rate – Sets the pointing rates.
Pt.Rate.inc – Increments the pointing rate.
Pt.Save – Add IH, ID correction to the MAdj registor and save the value to disk.
Pt.Sep - Specify the separation between pointing targets
Pt.Spiral.In – Command to spiral inwards.
Pr.Spiral.Out – Command to spiral outwards.
Pt.Spiral.Stop – Stop the spiral..
Pt.Spiral.center – Centers the pointing map spiral.
Pt.Spiral.rate – Controls the pointing map spiral’s speed.
Pt.Spiral.wid – Controls the pointing map spiral’s size.
ID, IH, NP, CH, ME, MA, HCES, HCEC, DCES, DCEC, FO TF, TX – Commands to modify pointmap coefficients using their tpoint names (See **Pt.Map.Set**)

2.6. Application Control and Misc

Die - Terminate the main IC process
Dome.Sim - Enable or disable dome control simulation mode
FioA.Sim - Enable or disable the simulation mode for the fio_a processes
FioB.Sim - Enable or disable the simulation mode for the fio_b processes
Fioc.Env.Update - Enable or disable fio_c updating of vtcs environment variables
FioC.Sim - Enable or disable the simulation mode for the fio_c processes
FioD.Sim - Enable or disable the simulation mode for the fio_d processes
FioE.sim – Enable or diable the simulation mode for the fio_e process.
FioF.sim - Enable or disable the simulation mode for the fio_f processes
FioMC.sim- Enable or disable the simulation mode for the fio_mc process.
FioHexe.sim – Enable or disables the simulation mode for the fio_hexe process.
FioDome.Sim - Enable or disable the simulation mode for the fio_dome processes
Help - Print a list of available tcs3 commands
Hist – Enable/Disable recording of history data.
Hist.Dir - Set the directory path for the history file
Log - Log a message to the log file and XUI display(s)
Log.Err - Log message to log file, XUI display(s), and error log
Logxui - Log message to the XUI display(s)
Notice.Print – ON to enable print statements when a warning/error is played in IC xterm.
Notice.Text - Enable/disable display of an error or warning notice in the mcc 'Warnings' window
Notice.Sound - Enable/disable playing of the sound associated with a notice
Pstart - Start or kill and restart child task
Pstop - Kill a child application

3. TCS3 Commands Details

This section describes the full set of TCS3 commands, with syntax and an example for each.

Add.Offset.Opt – How to add offsets to the base position.

```
  Syntax:   Ape.Mode {none|cosdec}
            none – Simple addition, target = base + offset.
            cosdec – Apply cos(dec) to offset:
                 target = base + (cos(dec)*offset).
 Example:   Set to cosdec option
            Add.Offset.Opt cosdec
```

Ape.Mode – Specifiy the mode for the fio_ape program. This program obtains the APE data for tcs3.

```
  Syntax:   Ape.Mode {off|on|sim|simtac}
            Off – No update are performed.
            On – Obtains APE data from APE hardware.
            Sim – Simulate data using vtcs mount(ha,dec) position.
            Simtac – read data from simtac lab computer.
 Example:   Set the mode to on
            Ape.mode on
```

Ape.Pos – Manual method for initialing the APE value. In case of hardware failure you can set the APE using the following sequence of commands:

```
            Kill fio_ape         Kill automatic apes update
            Ape.pos HA DEC       Manually set ape position
            Ape.set.apos         Initialize the APE position
  Syntax:   Ape.Pos HA DEC
 Example:   TO set the APE data pos to zenith:
            Ape.pos 00:00:00 19:49:34.39
```

Ape.set.pmac – Tells the tcs to initialize the position in the PMAC servo controller.

```
  Syntax:   ape.set.pmac
 Example:   ape.set.pmac
```

Autopid – Enable/Disable the AutoPID feature. AutoPID allow you to auto load different PID value during tracking and slewing..

```
  Syntax:   autopid {off|on}
 Example:   autopid off
```

Autopid.Set – Specifies a set of auto PID coefficients.

```
  Syntax:   autopid.set  ID P I D
            ID – Identifies the PID set. Vaild names are: track.ha,
            track.dec, slew.ha, slew.dec.
            P I D – Numeric PID values. Ranges is -100000 to 100000.
 Example:   autopid track.ha 40000 75000 6000
```

Base – Load a new position into the position table base.

Syntax: `base RA` *`Dec pm_ra pm_dec epoch equinox CS Name`*

`RA` - RA as time: hh:mm:ss.ss

`Dec` - Dec as angle: deg:mm:ss.ss

`pm_ra` - proper motion as sec(tm)/year. Optional, default = 0. In dRa/dt rather than cos(Dec)*dRA/dt.

`pm_dec` - proper motion as arcsec(tm)/year. Optional, default = 0

`epoch` – epoch in calendar years. Optional, Defaulted to ptable value.

`equinox` – equinox in calendar years Optional, Defaulted to ptable value.

`CS` - Coordinate system. Can be { FK5 | FK4 | APP }. Defaults to ptable value.

`Name` – Object name.

Example: Set the base position to the object SAO-93498:

```
Base 0:34:56.51 19:48:36.1 0.0011 -0.0270 200 2000 FK5 SAO-93498
```

Base.inc – Increment the RA, Dec values of the base.

Syntax: `base.inc` *`ra dec`*

`Ra dec` – value to increment in arcseconds.

Example: `Base.inc 0 1`

Beam.Init– Initializes the Beam offset by: making the current position the base (0,0), setting the ra, dec values to 0,0, and enabling the offset.

Syntax: `Beam.init`

Example: Initialize the Beam offsets

```
Beam.init
```

Beam.inc – Apply an increment to the beam RA and Dec offsets

Syntax: `beam.inc` *`ra dec`*

`Ra dec` – Offset increments in arcseconds

Example: Increment beam Dec offset by 1.8 arcseconds, leave RA offset as-is

```
Beam.inc 0 1.8
```

Beam.off – Disable application of the beam RA and Dec offsets

Syntax: `beam.off`

Example: Disable application of the current beam RA and Dec offsets

```
Beam.off
```

Beam.on – Enable application of the beam RA and Dec offsets

Syntax: `beam.on`

Example: Apply the current beam RA and Dec offset values

```
Beam.on
```

Beam.set – Specify the RA and Dec offsets for the beam

Syntax: `beam.set` *`ra dec`*

`Ra dec` – Offsets in arcseconds

Example: Set a beam offset of 10.2 arcseconds in RA, -6.5 arcseconds in Dec

```
Beam.set 10.2 -6.5
```

Beam.toggle – Toggle the beam offset enable status

Syntax: `beam.toggle`

Example: Toggle to alternate beam position

```
Beam.toggle
```

Cat.index – Search specified catalog by index and load results into ‘next’ buffer

Syntax: `cat.index` *`CAT Index`*

`CAT` - Catalog name (bsc5, fk5, gsc, text, sao, ukirt, hd_sao)

`Index` – catalog index of object to search for

Example: Load next buffer with GSC object with index 119001564

```
Cat.Index gsc 119001564
```

Cat.search – Search catalog for the star closest to given RA/Dec within a specified radius

Syntax: `Cat.search` *`CAT RA Dec Radius`*

`CAT` - Catalog name (bsc5, fk5, gsc, text, sao, ukirt, hd_sao, fk5m)

`RA` - RA as time: hh:mm:ss.ss

`Dec` - Dec as angle: deg:mm:ss.ss

`Radius` – Search radius around target RA/Dec in arcseconds

Example: Search GSC catalog for guide star within 200 arcsecond radius of object SAO-93498

```
Cat.search gsc 0:34:56.51 19:48:36.1 200
```

Collimate.enable – Enable or disable collimation control loop

Syntax: `collimate.enable` *`control`*

`control` – control command (OFF or ON)

Example: Turn collimation control loop off

```
Collimate.enable off
```

Collimate.EW.Dpos– Set desired East/West collimation position

Syntax: `collimate.ew.dpos` *`dpos`*

`dpos` – Target E/W collimation position as voltage Range is -4.80 to 8.08 volts for chopper, -43.63 to +43.63 for the Hexapod.

Example: Set East/West collimation position to 5.4 volts

```
Collimate.EW.Dpos 5.4
```

Collimate.EW.Dpos.Inc – Increments East/West collimation’s desired position

Syntax: `collimate.ew.dpos.inc` *`inc`*

`inc` – Incremment value.

Example: Increments East/West collimation position to 0.1 units.

```
Collimate.EW.Dpos.Inc 0.1
```

Collimate.NS.Dpos – Set desired North/South collimation position

Syntax: `collimate.ns.dpos` *`dpos`*

`dpos` – Target N/S collimation position as voltage voltage Range is -9.0 to 4.06 volts for chopper, -43.63 to +43.63 for the Hexapod.

Example: Set North/South collimation position to -3.6 .

```
Collimate.NS.Dpos -3.6
```

Collimate.NS.Dpos.Inc – Increments North/South collimation's desired position

Syntax: `collimate.ns.dpos.inc` *`inc`*

`inc` – Incremment value.

Example: Increments North/South collimation position to 0.1 units.

```
Collimate.NS.Dpos.Inc 0.1
```

Collimate.Table.Read – Reads the file ~/data/collimate.txt and loads collimation table data into TCS3. As of 2/2020, default collimation and focus are provide by the collimation table.

Syntax: `collimate.table.read`

Example:

```
Collimate.table.read
```

Collimate.Table.Set – Sets the collimation's desired position using *name* and the collimation table data.

Syntax: `collimate.table.set` *`name`*

`name` – A text name of an entry in the collimation data.

Example: Set the collimation for spex

```
Collimate.table.set spex
```

CS – Set the default coordinate system

Syntax: `CS` *`coord_sys`*

`Coord_sys` – Coordinate system (fk4, fk5, app)

If using App, remember to set the ptable epoch to the current jepoch to 'time stamp' the apparent coordinates to the current date.

Example: Set default coordinate system to fk5

```
CS fk5
```

CW.Dir – Set the directory path for counterweight files

Syntax: `cw.dir` *`path`*

`path` – full path name (defaults to /home/tcs3/data/cw)

Example: Set counterweight path to /home/tcs3/data/cw_041214

```
CW.dir cw.dir /home/tcs3/data/cw_041214
```

CW.Dpos– Set selected counterweight to position described as a voltage

Syntax: `cw.dpos` *`name position`*

`name` – Name of the counterweight to be positioned (e.g. TV2S)

`position` – Desired counterweight position as voltage. If the voltage is outside the min/max range of the counterweight, it will be clipped to its min/max value.

Example: Move counterweight YH3F to position 4.23 volts

```
CW.dpos yh3f 4.23
```

CW.Enable – Enable or disable counterweight movements

Syntax: `cw.enable` *`control`*

`control` – Enable control (ON or OFF)

Example: Enable control of the counterweight system
`CW.enable on`

CW.Manual.cntl – The CW command to set a counterweight up, down or off
Syntax: `cw.manual.cntl` *`control`*
`control` – { off | 01.up | 01.dn | 02.up | 02.dn | .. | 10_11.up | 10_11.dn)
Example: Enable counterweight manual control mode
`CW.manual.cntl on`

CW.Mode – Select the counterweight operational mode
Syntax: `cw.mode` *`mode`*
`Mode` – operational mode (Auto | Manual | Lock)
Example: Place counterweights in manual mode
`CW.mode manual`

DHP.Enable – Enable/Esiable Dome Hand Paddle inputs.
Syntax: `DHP.enable` *`{off|on}`*
`Off` – ignores the dome handpaddle IO
`On` – accepts dome handpaddle inputs
Example: Enable the dome handpaddle.
`DHP.enable on`

DHP.Rate – Set the maximum velocity for the dome handpaddle.
Syntax: `DHP.rate` *`rate`*
`rate` – This value represents arcseconds/seconds. Range is 0 to 800 as/s.
Example: Set the rate to 200 as/s.
`DHP.rate 200`

DHP.swapEW – Swap the logic for the East and West buttons on the dome handpaddle.
Syntax: `DHP.swapEW` *`{off|on}`*
`Off` – East is east, …
`On` – East is west, …
Example: Set the default mapping
`DHP.swapEW off`

DHP.swapNS – Swap the logic for the North and South buttons on the dome handpaddle.
Syntax: `DHP.swapNS` *`{off|on}`*
`Off` – North is north, …
`On` – North is south, …
Example: Set the default mapping
`DHP.swapNS off`

Die – Terminate the main IC process
Syntax: `die`
Example: Terminate the current ic process

```
Die
```

Dome.Auto.Offset –Specifies an offset angel to be used during Dome Control's auto (tracking) mode.

Syntax: `dome.auto.offset` *`deg`*

`deg` `- offset in degrees (-180 to 180).`

Example: Set the auto offset value to 20 degrees:

```
Dome.auto.offset 20
```

Dome.Capture –Capture some dome serial data to a file. Debugging command.

Syntax: `dome.capture`

Example: Start the dome capture:

```
Dome.capture
```

Dome.Goto –Specifies a dome position for goto mode.

Syntax: `dome.goto` *`AZ_des`*

`AZ_deg` `- azimuth in degrees (0-360).`

Example: Moved the dome to a azimuth of 90 degrees:

```
Dome.goto 90
```

Dome.HP.Speed – Sets the maximum speed in dome handpaddle mode.

Syntax: `dome.HP.Speed` *`speed`*

`speed` – speed factor 0.0 to 1.0 maps to 0 to 10 volt max output to the Ampilifiers.

Example: Set the max speed to ½, or 0.5. Maxinum of 5 volt to the ampilifiers inputs.

```
Dome.HP.Speed 0.5
```

Dome.Manual –Control dome motion while in manual mode

Syntax: `dome.manual` *`motion`*

`motion` – Desired dome motion (forward, reverse, stop)

Example: Move dome in the reverse direction

```
Dome.manual reverse
```

Dome.Manual.Speed – Sets the maximum speed in dome manual mode.

Syntax: `dome.Manual.Speed` *`speed`*

`speed` – speed factor 0.0 to 1.0 maps to 0 to 10 volt max output to the Ampilifiers.

Example: Set the max speed to ½, or 0.5. Maxinum of 5 volt to the ampilifiers inputs.

```
Dome.Manual.Speed 0.5
```

Dome.Mode – Set the dome movement control mode

Syntax: `dome.manual` *`mode`*

`mode` – Dome movement control mode (auto | manual | lock)

Example: Put dome in manual control mode

```
Dome.mode manual
```

Dome.Sim – Enable or disable dome simulation mode

Syntax: `dome.sim` *`control`*

`control` – Dome simulation mode (ON or OFF)

Example: Turn dome simulation off

`Dome.sim off`

Dome.Speed – Set the dome movement speed factor

Syntax: `dome.speed` *`speed`*

`speed` – Speed as a factor of full speed (0.0 to 1.0)

Example: Set dome speed to half of full speed

`Dome.speed 0.5`

Elevation – Set the elevation of the observatory

Syntax: `elevation` *`meters`*

`meters` – Elevation of the observatory above sea level (meters)

Example: Set the elevation to 4168 meters above sea level

`Elevation 4168`

Epoch – Set the default epoch

Syntax: `epoch` *`year`*

`year` – Epoch of the CS as a calendar year (1900 to 2100)

Example: Set the default epoch to 1950

`Epoch 1950`

Equinox – Set the default equinox

Syntax: `equinox` *`year`*

`year` – Equinox as a calendar year (1900 to 2100)

Example: Set the default equinox value to 2000

`Equinox 2000`

FioA.Sim – Enable or disable the simulation mode for the fio_a processes

Syntax: `fioa.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode off for fio_a processes

`Fioa.sim off`

FioB.Sim – Enable or disable the simulation mode for the fio_b processes

Syntax: `fiob.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode on for fio_b processes

`Fiob.sim on`

FioC.Sim – Enable or disable the simulation mode for the fio_c processes

Syntax: `fioc.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode off for fio_c processes

`Fioc.sim off`

FioDome.Sim – Enable or disable the simulation mode for the fio_dome processes

Syntax: `fiodome.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode off for fio_dome processes

```
Fiodome.sim off
```

FioD.Sim – Enable or disable the simulation mode for the fio_d processes

Syntax: `fiod.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode on for fio_d processes

```
Fiod.sim off
```

FioE.Sim – Enable or disable the simulation mode for the fio_e processes

Syntax: `fioe.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode on for fio_e processes

```
Fioe.sim on
```

FioF.Sim – Enable or disable the simulation mode for the fio_f processes

Syntax: `fiof.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode on for fio_f processes

```
Fioe.sim on
```

FioHexe.Sim – Enable or disable the simulation mode for the fio_hexe processes

Syntax: `fioHexe.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode on for fio_hexe processes

```
FioHexe.sim on
```

FioMC.Sim – Enable or disable the simulation mode for the fio_mc processes

Syntax: `fioMC.sim` *`control`*

`control` – Simulation mode control (OFF or ON)

Example: Turn the simulation mode on for fio_mc processes

```
FioMC.sim on
```

Focus.Adj.Enable – Enable or disable the focus adjustment value. A focus adjustment value is calculated by the TCS based on the telescope truss temperature and it position. These values are applied to the TCS in real-time when adj.enable is ON.

Syntax: `focus.adj.enable` *`{ OFF | ON }`*

`OFF` – ignore the focus adjustment values. Dpos = user_dpos.

`ON` – apply adjustment value, dpos = user_dpos + adj.

Example: Turn on focus adjustment:

```
Focus.adj.enable on
```

Focus.Dpos – Set the desired focus position

Syntax: `focus.dpos` *`position`*

`position` – Desired focus position as voltage Range is -7.33 to 7.00 volts for the chopper, and -8 to 8 mm for the hexapod.

Example: Set the focus position to 2.4.

```
Focus.dpos 2.4
```

Focus.Dpos.Inc – Increment or decrement the focus position

Syntax: `focus.dpos.inc` *`pos_inc`*

`Pos_inc` – Position increment/decrement value.

Example: Decrements the focus position by 0.43.

```
Focus.dpos.inc -0.43
```

Focus.Enable – Enable or disable the focus control loop

Syntax: `focus.enable` *`control`*

`control` – Focus loop control (ON or OFF)

Example: Disable the focus control loop

```
Focus.enable off
```

Help – Print a list of available tcs3 commands

Syntax: `help`

Example: Print the list of commands

```
Help
```

Hexapod.Init – Will cause the TCS to run the hexapod init macro /home/tcs3/data/hexapod_init. TCS3 has limited feedback, so it is suggested the the hexegui be used to initialize the hexapod.

Syntax: `Hexapod.init`

Example: `Hexapod.init`

Hist – Enable/Disable the collection of the realtime history data. Off is useful to freeze the graphical dislay on the mcc to view the data.

Syntax: `Hist` *`{ off | on }`*

`on` – Collect the data.

`off` – Stop data collections.

Example: `Hist on`

Hist.Dir – Set the directory path for the history file

Syntax: `hist.dir` *`path`*

`path` – Directory path for the history file (defaults to /home/tcs3/data/hist)

Example: Set the history file directory path to /home/tcs3/data/hist

```
Hist.dir /home/tcs3/data/hist
```

HSlew.limit.override – Override the Horizontal Slew Limits. This command affect the software only.

Syntax: `HSlew.limit.override` *`{ off | on }`*

`on` – Override the limits

`off` – Slow down the max velocity to 400 as/s, if limit are trigged.

Example: Disable HSlew limit

```
HSlew.limit.override on
```

Humidity– Manually set the relative humidity

Syntax: `humidity` *`value`*

`value` – Relative humidity value (0.0 to 1.0)

Example: Set the relative humidity to 0.56

```
Humidity 0.56
```

Humidity.Wn – Set relative humidity level to trigger warning.

Syntax: `humidity.wn` *`value`*

`value` – Trigger a warning if relative humidity exceeds this value (0.0 – 1.0)

Example: Set the relative humidity warning level at 80%

```
Humidity.wn 0.8
```

Stop – Put the tcs system in the ‘Stop mode

Syntax: `Stop`

Example: Puts the TCS3 Servo in the stop mode. In Stop, the pmac is in open loop, telescope brakes are on.

```
Stop
```

Info – Return a selected subset of tcs information

Syntax: `info` *`selection`*

`selection` – The data to display using these identifiers: TM TMr SP SPr MP MPr OS OP OPr SM FO CO EN US DO PM ON NS WE SH MC GI SI WE1

The format of each selection are described below. Multiple selections can be made, for example: `info TM OP`

The format for each parameter is described below:

TM = UTC(yyyy/mm/dd) UTC(hh:mm:ss.ss) UTC_local(yyyy/mm/dd) UTC_local(hh:mm:ss.ss) Last(hh:mm:ss.ss)

TMr = UTC(mdj) UTC_local(mjd) Last(radians)

UTC is coordinated universal time.

UTC_local is the local time (HST).

Last is Local Apparent Sideral Time.

SP = target_ra(hh:mm:ss) Target_da(deg:mm:s) pm_ra(sec/year) pm_dec(arcsec/year) epoch(yyyy.y) equinox(yyyy.y) CS(string)

SPr = target_ra(radians) Target_dec(radinas) pm_ra(sec/year) pm_dec(arcsec/year) epoch(yyyy.y) equinox(yyyy.y) CS(string)

The target RA, Dec is where the telescope is pointing (Base position + any offset).

Pm_ra/dec is the proper motion data.

Epoch is the epoch of the RA, Dec coordinates.

Equinox – is the epoch of the coordinate system.
CS – identifies the coordinate system (fk5, fk4, app).

MP = mean_ra(hh:mm:ss) mean_da(deg:mm:s)
MPr = mean_ra(radians) mean_da(radians)

Mean_ra, dec – The mean (fk5, J2000 equinox, current epoch) RA and DEC.

OS = TotalOS(ra dec) UserOS(ra dec enable) BeamOS(ra dec enable) ScanOS(ra dec) [all in arcsec]

TotalOS – Total offset value.
UserOS – User offset magnitude of RA and DEC in arcseconds. Enable=1 of apply, 0 if ignored.
BeamOS – Beam offset magnitude of RA and DEC in arcseconds. Enable=1 of apply, 0 if ignored.
ScanOS – Scan offset magniture in RA and DEC.

OP = obs_ra(hh:mm:ss) obs_dec(deg:mm:s) obs_ha(hh:mm:ss) Am(1.00) Zn(deg) azimuth(deg) PA(deg)
OPr = obs_ra(radians) obs_dec(radians) obs_ha(radians) Am(1.00) Zn(deg) azimuth(deg) PA(deg)

Obs_ra, obs_dec, obs_ha – The observered RA and DEC, and Hour Angle coordinates.
Am is Airmass.
Zn is Zenith distance.
Azimuth is the telescope azimuth position.
PA is the parallatic angle.

SM = mode (track/slew/...) value

Mode is the current servo mode of the tcs3. They are: Track, slew, MV, MP, stop.
Value is data related to a particulare servo mode, for slew and MP the estimated completion time in seconds of the current move is retuned. All other mode return 0.

FO = Focus.dpos focus.apos focus.enable

Focus.dpos is the user requested focus position.
Focus.apos is the actual measured focus position.
Focus.enable is a flag (0 for OFF, 1 for ON) to indicate if remote focus commands are accepted by tcs3.

FO2 = Foscus.user_dpos Focus.adjustment Focus.dpos focus.apos focus.enable focus.adj.enable

Focus.user_dpos is the user requested focus position.
Focus.adjustment is the Temperature and Position adjustment value calculated by the TCS.
Focus.dpos is the desired position for the focus mechanism. dpos = user_dpos + adjust (if adj.enable is ON).
Focus.apos is the actual measured focus position.
Focus.enable is a flag (0 for OFF, 1 for ON) to indicate the software control loop is active. When OFF, focus position commands are not accepted by the TCS.
Focus.adj.enable flag is 0, or OFF when the adjustment value is ignore. When 1, or ON, the adjust value is added to user_dpos to get dpos.

CO = Coll_EW.dpos Coll_EW.apos Coll_NS.dpos Coll_NS.apos Collimation.enable

Coll_EW.dpos is the collimation EW desired position in volts.
Coll_EW.apos is the collimation EW actual position in volts.
Coll_NS.dpos is the collimation NS desired position in volts.
Coll_NS.apos is the collimation NS actual position in volts.
Collimation.enable is a flag (0 for OFF, 1 for ON) to indicate if collimation control is enable.

EN = Encoder data: motor_counts.ha(counts) motor_counts.dec(counts) apes.pos.ha(radians) apes.pos.dec(radians)

Motor_counts ha dec – Motor position in counts are reported by the Motor Controller Hardware (PMAC).
APE_pos ha dec – The Absolute Position Encoder position are reported by the APE Hardware.

US = User Spiral data: state position rate width

State: -1 for down, 0 for stop, 1 for up.
Position: position along the spiral where 2*PI is 1 rotation out from center.
Rate: Rate of speed while moving alow the spiral in as/s.
Width: Width of 1 rotation, ie at position 6.28 the xy offset would be [1*width, 0]

DO = Dome Info: az(deg) vel(deg/s)

Az: dome azimuth in degrees.
Vel: dome velocity in arcseconds/seconds.

PM = Pointing Map information: Map.IH(as) Map.ID(as) madj.IH(as) madj.ID(as) peak.IH(as) peak.ID(as)
Sp.Rot Sp.IH(as) Sp.ID(as) Rates.pos.IH(as) Rate.pos.ID(as) Rate.vector.IH(as/s) Rate.vector.ID(as/s)

Map.IH/ID – The IH/ID values of the point map in arcseconds.
madj.IH/ID – The IH/ID values in the Map Adjustment registers, arcseconds.
peak.IH/ID – The IH/ID values in the Peak Adjustment registers, arcseconds.
Sp.Rot – Number of rotation for the spiral.
Sp.IH/ID – The IH/ID values for the Sprial registers, arcseconds.
Rates.pos.IH/ID – The IH/ID values from the pointing rates registers, arcseconds.
Rate.vector.IH/ID – The rates in the pointer rates register, arcseconds per seconds.

ON = Object_name Object_magnitude
ONS = Object_name Object_magnitude Object_source

Object_name: The object name in the ptable.
Object_magnitude: the magnitude from the ptable.
Object_source: Source for the information

NS = Non-sidereal rates: ra(as/s) dec(as/s)

Ra dec: the non-sidereal rates in the ptable.

WE = Weather: air_temperature(c) humidity(0-100) wind_speed(mph) Wind_dir(deg) dome_td6(c)

Air_temperature in celcius.
Humidity, 0 to 100.
Mean_Wind_speed West Sensor, mph
Wind_direction West Sensor, degrees.
Dome Temperature from sensor TD6 in celcius.

SH = sh_ulimit sh_touch sh_dlimit sh_block_time(hr)

Sh_ulimit – upper limit state of shutter, 0 or 1.
Sh_touch – The touch limit state, 0 or 1.
Sh_dlimit – The lower shutter down limit state, 0 or 1.
Block_time – Number of hours until view of field (while tracking) with be blocked by the shutter, hours.

MC = mc.air.in(degC) mc.air.out(degC) mc.glycol(degC) actuator(v) fan(0/1) heat(0/1) mc.pressure_err(0/1)

mc.air.in – Intake air temperature for mirror cooling in degrees C.
mc.air.out – Outtake air temperature for mirror cooling in degrees C.
mc.glycol – Glycol temperature used by mirror cooling in degrees C.
actuator – control value for mc.actuator command in volts.
Fan – control value for mc.fan command, 0 is OFF, 1 is ON.
Heat – control value for mc.heater command, 0 is OFF, 1 is ON.
mc.pressure_err – Mirror Cooling Pressure Error value , 0=no error, 1= pressure error.

SI = Secondary_information

Secondary_information: 0=chopper, 1=hexapod

GI = on-axis-mirror_state pt_peak_commands

On-axis-mirror – 1=IN, 0=OUT, -1=UnKnown
Pt_peak_commands = number of pt_peaks_* command over the last 60 sec. These are guider command send to the TCS.

WE1 = hmt300-humitidy(0-100) hm300-air_temp-degC hmt300.DewPt-degC

hmt300-humitidy(0-100) – HMT 300 humidity 1 to 100.
hm300-air_temp-degC - HMT 300 Air Temperature in DegC.
hmt300.DewPt-degC – HMT 300 Dew Point Temperature in DegC.

WE2 = BP MWSE WDE 0.0 0.0

BP – Pressure in mbar

MWSE – Mean Wind Speed, East sensor in MPH
WDE – Wind Direction, East Sensor in degrees.
2 zero values to follow. Reserved for future value.

MCS = mirror_cover mirror_support.offon mirror_support.volts mirror_support.fault_latch

Mirror_cover: 0=shut, 1=open
Mirror_support.offon: 0=off, 1=on
Mirror_support.volts: voltage representing mirror support air pressure.
Mirror_support.fault_latch: 0=OK, 1=FAULT

LS = Last Slew: valid cs eq ra(rad) dec(rad) pm.ra(s/y) pm.dec(as/y) pm.ep nsrate.ra(as/s) nsrate.dec(as/s) name source

valid: 0=data not valid; 1=valid
cs: coordinate system, ie “FK5”
eq: equinox of cs.
ra dec: ra and dec location in radans
pm.ra dec: proper motion
ep: epoch for the proper motion.
nsrate.ra dec: non-sidereal rate of the object.
name - name of the object
src – source of the information.

SE1 = Seeing data 1: DIMM_valid DIMM MASS_valid MASS TAU_valid TAU225GHZ
DIMM_valid: 0=not valid, 1=valid
DIMM: DIMM seeing value
MASS_valid: 0=not valie, 1=valid
MASS: MASS seeing value
TAU_valid: 0=not valid, 1=valid
TAU225GHZ – TAU seeing value.

Example: An example of each option is display below:

```
info TM
OK 2008-02-08 23:44:57.204 2008-02-08 13:44:57.20 22:36:54.44

info TMr
OK 54504.9899559505 54504.5732892839 5.923176443112

info SP
OK  22:36:25.63  19:46:07.7 0.0000 0.0000 2000.0 2000.0 fk5

info SPr
OK 5.91873876393661 0.34503048854095 0.0000 0.0000 2000.0 2000.0 fk5
```

```
info MP
OK  22:36:33.03  19:46:07.6

info MPr
OK 5.91925801245125 0.34503030521678

info OS
OK 0.0 0.0 0.0 0.0 0 0.0 0.0 0 0.0 0.0

info OP
OK  22:37:09.28  19:48:37.2  00:00:47.77 1.000 0.188 265.185 -85.120

info OPr
OK 5.92218610367147 0.34575561422440 0.00347363680141 1.000 0.188 265.185 -
85.120

info SM
OK stop 0

info FO
OK  -0.340 -0.338 0

Info FO2
OK 0.100 0.082 0.182 0.180 1 1

info CO
OK -2.71 -2.71 -2.15 -2.15 1

info EN
OK  13923 1426743 0.00199361128469 0.34680085134736

Info US
OK 0 0.00 20.0 60.0

Info DO
OK  116.6 2.0

Info PM
OK  91.1 -96.1 -11.0 -21.0 3.0 2.0 0.00 -0.0 0.0 0.0 0.0 0.0000 0.0000

Info ON
OK no_name 0.0

Info NS
OK 0.0000 0.0000

Info WE
OK 2.0 20.1 12.3 157.0

Info SH
OK 0 1 1 3.0000

Info SH
OK 0 1 1 2.9667

Info MC
OK 4.2 1.2 1.0 0.0 1 0
```

```
Info SI
OK 1

Info GI
OK 0 0


Info WE1
OK 10.5 10.9 -19.2

info we2
OK 620.80 15.4 57.0 0.0 0.0

info mcs
OK 0 0 -0.01 0

info ls
OK 0 fk5 2000.0 4.65260180469594 0.34603280752848 0.000000 0.000000 2000.000
0.000000 0.000000 noname unknown

info se1
OK 1 0.440 1 0.250 0 0.000
```

JS.Enable – Enable/Diable TO Panel Joystick inputs.

Syntax: `JS.enable` *`{off|on}`*

`Off` – disable the joystick inputs

`On` – enables the joystick inputs

Example: To enable the Joystick.

`JS.enable on`

JS.Rate – Set the maximum velocity for the TO Panel Joystick input.

Syntax: `JS.rate` *`rate`*

`rate` – This value represents arcseconds/seconds. Range is 0 to 800 as/s. When the tcs3, is in track mode it will limit the rate to under 60 as/s.

Example: Set the rate to 10 as/s.

`JS.rate 10`

JS.swapEW – Swap the logic for the East and West inputs on the TO Panel Joystick.

Syntax: `JS.swapEW` *`{off|on}`*

`Off` – East is east, …

`On` – East is west, …

Example: Set the default mapping

`JS.swapEW off`

JS.swapNS – Swap the logic for the North and South buttons on the TO Panel Joystick.

Syntax: `OH1.swapNS` *`{off|on}`*

`Off` – North is north, …

`On` – North is south, …
Example: Set the default mapping
`JS.swapNS off`

Latitude – Set the latitude of the observatory
Syntax: `latitude` *`latitude hemisphere`*
`latitude` – Latitude in deg:mm:ss (must be <= 90)
`hemisphere` – Hemisphere of the observatory (N or S)
Example: Set to 19 deg, 49 min, 34.39 sec north latitude
`Latitude 19:49:34.39 N`

Log – Log a message to the log file and XUI display(s)
Syntax: `log` *`message`*
`message` – Text of the message to be logged
Example: Log the message ‘object obscured by upper level clouds’
`Log object obscured by upper level clouds`

Log.Err – Log message to log file, XUI display(s), and error log
Syntax: `log.err` *`message`*
`message` – Text of the message to be logged
Example: Log the error message ‘unable to acquire outside air temperature’
`Log.err unable to acquire outside air temperature`

Log.xui – Log message to the XUI display(s)
Syntax: `logxui` *`message`*
`message` – Text of the message to be logged
Example: Display message ‘starting run’ on all XUI displays
`Log.xui starting run`

Longitude – Sets the telescope’s longitude.
Syntax: `Longitude` *`d:m:s E|W`*
`d:m:s` – Longitude in deg, min, sec. Must be >= 180.
`E W` – Indicates East or West of Greenwich.
Example: The IRTF’s longitude:
`Longitude 155:28:19.2 W`

Ltc.Enable – Enable/disable Laser Traffic Control updates
Syntax: `ltc.enable` *`control`*
`control` – LTC update control (ON or OFF)
Example: Turn LTC updating off
`Ltc.enable off`

Ltc.Filename– Set the full path for the LTC file name
Syntax: `ltc.filename` *`filename`*
`filename` – LTC filename and path (defaults to /tmp/tcs_data.txt)

Example: Set the LTC file name to /tmp/ltc_data_041214.txt

```
Ltc.filename /tmp/ltc_data_041214.txt
```

Ltc.Fov – Set Laser Traffic Control Field of View

Syntax: `ltc.fov` *`fov`*

`fov` – LTC field of view in degrees (0.001 to 1.6666667) [100 arcmin]

Example: Set LTC field of view to 0.01 deg

```
Ltc.fov 0.01
```

Ltc.Impact – Set the value for the Laser Traffic Control ‘impact’ field

Syntax: `ltc.impact` *`value`*

`value` – Value for LTC impact field (YES or NO)

Example: Set LTC impact field to NO

```
Ltc.impact no
```

Ltc.Period – Set period between LTC file updates

Syntax: `ltc.period` *`time`*

`time` – Time between file updates in seconds (5 to 300)

Example: Set LTC file update period to 60 seconds

```
Ltc.period 60
```

LTOffset – Set the ut to local time offset

Syntax: `ltoffset` *`time`*

`time` – Local time offset in minutes (-720 to 720)

Example: Set local time offset to – 600 minutes (-10 hours west)

```
Ltoffset -600
```

MC.Actuator – Sets the mirror cooling actuator voltage command used to control the glycol flow.

Syntax: `mc.actuator` *`volts`*

`Volts –` Control voltage for actuator. Range is 0 to 5.0 volts for close to fully open.

Example: Close the actuator, or turns off cooling.

```
mc.actuator 0
```

MC.fan – Sets the mirror cooling Fan control output to OFF or ON.

Syntax: `mc.Fan` *`{ off | on }`*

Example: Turning the fan on.

```
mc.fan on
```

MC.Heater – Sets the mirror cooling Heater control output to OFF or ON.

Syntax: `mc.heater` *`{ off | on }`*

Example: Turning the heater off.

```
mc.heater off
```

MC.Mode – Sets the mirror cooling control mode. The control mode tells the TCS how to control the mirror cooling

Syntax: `mc.mode` *`{ off | manual | auto }`*

off – Off turns the Fan OFF, Heater OFF, and Actuator to 0v.
manual – Allow the user to control the fan, heater, and actuator manually.
auto – TCS control mirroring cooling.

Example: Tuning of mirror cooling off.

```
mc.mode off
```

MC.SetPt – Sets the mirror cooling Set Point value in degrees C. The set point is the temperature you wish to cool the mirror to.

Syntax: `mc.setpt` *DegC*

Example: Enter a set point of 1 degree C.

```
mc.setpt 1
```

Mirror.Cover – Open or closes the mirror covers.

Syntax: `mirror.cover` *{ shut | open }*

Example: Close the mirror cover

```
Mirror.cover shut
```

Mirror.Support – Turns off/on the mirror support

Syntax: `Mirror.support` *{OFF | ON }*

Example: To turn mirror support on

```
Mirror.support on
```

MP – Execute a Motor Position move

Syntax: `mp` *HA Dec*
HA - HA as time: hh:mm:ss.ss
Dec - Dec as angle: deg:mm:ss.ss

Example: Do a motor position to the zenith

```
Mp 00:00:00 19:49:34.39
```

MP.inc – Increments the MP destination position in arcseconds.

Syntax: `mp.inc` *HA Dec*
HA - HA in arcseconds.
Dec - Dec in arcseconds

Example: Increments HA by 10, and dec by 20 arcseconds.

```
Mp.inc 10 20
```

MP.cnt – Specity the MP destination position in motor counts.

Syntax: `mp.cnt` *HA Dec*
HA - HA in motor counts.
Dec - Dec in motor counts

Example: Move to HA 0, Dec 0.

```
Mp.cnt 0 0
```

MP.Vel – Set the velocity for a motor position move

Syntax: `mp.vel` *velocity*
velocity – Velocity in as/s (1 to 1600)

Example: Set MP velocity to 400 as/s

```
Mp.vel 400
```

MS.Fault.Ack – Re-enable mirror support after a fault

Syntax: `ms.fault.ack`

Example: Re-enable mirror support after a fault

```
Ms.fault.ack
```

MV – Execute a Motor Velocity move

Syntax: `mv` *`HA Dec`*

`HA` - HA as time: hh:mm:ss.ss

`Dec` - Dec as angle: deg:mm:ss.ss

Example: Do a motor velocity move to the zenith

```
 Mv 00:00:00 19:49:34.39
```

MW.Zero – Zeros the value of the measurement widget. The measure widget is a simple tool to allow the TO to make a distance measurement by moving the telescope using the Offsets (User, beam) or the IH, ID correction registers. Theses offsets are added to the measure widget, and are display in the tracking status area on MCC1. This command allow the TO to zero the values prior to any measurement.

Syntax: `mw.zero`

Example: `mw.zero`

Next – Enter object data into the 'next' buffer

Syntax: `next RA` *`Dec pm_ra pm_dec epoch equinox CS Name Mag NS_rate_RA NS_rate_Dec`*

`RA` - RA as time: hh:mm:ss.ss

`Dec` - Dec as angle: deg:mm:ss.ss

`pm_ra` - proper motion as sec(tm)/year. Optional, default = 0. In dRa/dt rather than cos(Dec)*dRA/dt.

`pm_dec` - proper motion as arcsec(tm)/year. Optional, default = 0

`epoch` – epoch in calendar years. Epoch used for proper motion correction. Optional, Defaulted to ptable value.

`equinox` – equinox of coordinate system, in calendar years. Optional, defaults to ptable value.

`CS` - Coordinate system. Can be { FK5 | FK4 | APP }. If not specifed, defaults to current ptable value. App is Topocentric apparent.

`Name` – Object name.

`Mag` – Magnitute of object.

`Ns_rate_ra` – RA non-sidereal rate in as/s.

`NS_rate_dec` – DEC non-sidereal reate in as/s.

When CS=FK5, the default epoch and equinox is 2000.0.
When CS=FK4, the default epoch and equinox is 1950.0.
When CS=App, the following field are ignored: pm_ra pm_dec, equinox and epoch.

Example: Set the star catalog next buffer to the object SAO-93498:

```
Next 0:34:56.51 19:48:36.1 0.0011 -0.0270 2000 2000 FK5 SAO-93498 8.8 0.0 0.0
```

Next.HADec – object data into the 'next' buffer using HA and Dec coordinates

Syntax: `next HA Dec`

HA - HA as time: hh:mm:ss.ss

Dec - Dec as angle: deg:mm:ss.ss

Example: Setup the next object to zenith, or 0 Hours Ha 19:50 degrees Dec.

```
Next.hadec 0 19:50
```

Next.Clear – Clear a next star entry

Syntax: `next.clear` *entry*

entry – Flag to select specific next entry (to, ob, sc)

Example: Clear the next observer object entry

```
Next.clear ob
```

Notice.Print – When ON, the audio task prints the notices detected when playing the sound. This output can be viewed in the IC xterm.

Syntax: `notice.print` *[off | on]*

off – Do not print..

on – Display messages.

Example: `Notice.print off`

Notice.Sound – Enable/disable playing of the sound associated with a notice

[Not intended for general use – notice indices may change between system releases]

Syntax: `notice.sound` *index offon*

index – Specify index of the notice sound to be enabled or disabled.

offon – disable/enable notice sound (off | on)

Example: Disable the sounds associated with the humidity warning notice

```
Notice.sound 22 off
```

Notice.Text – Enable/disable display of an error or warning notice in the mcc 'Warnings' window

[Not intended for general use – notice indices may change between system releases]

Syntax: `notice.text` *index offon*

index – Specify index of the notice to be enabled or disabled.

offon – disable/enable text display (off | on)

Example: Disable display of the humidity warning text

```
Notice.text 22 off
```

NS.rate – Specifies a non-sidereal rate to be applied to the base position.

Syntax: `ns.rate` *ra dec*

Ra dec – Rate in arcseconds/second

Example: To cancel the earth's rotation during tracking:

```
ns.rate -15.0411 0
```

NS.rate.inc – Increments the base non-sidereal rate.

Syntax: `ns.rate.inc` *ra dec*

`Ra dec` – Add these values to the base rates in arcseconds/second

Example: `ns.rate.inc 0 1.3`

OH1.Enable – Enable/Diable Observers Hand Paddle #1 inputs.

Syntax: `OH1.enable {off|on}`

`Off` – ignores the OH1 handpaddle IO

`On` – accepts OH1 handpaddle inputs

Example: Enable the OH1 handpaddle.

`OH1.enable on`

OH1.Rate – Set the maximum velocity for the OH1 handpaddle.

Syntax: `OH1.rate rate`

`rate` – This value represents arcseconds/seconds. Range is 0 to 60 as/s.

Example: Set the rate to 10 as/s.

`OH1.rate 10`

OH1.swapEW – Swap the logic for the East and West buttons on the OH1 handpaddle.

Syntax: `OH1.swapEW {off|on}`

`Off` – East is east, …

`On` – East is west, …

Example: Set the default mapping

`OH1.swapEW off`

OH1.swapNS – Swap the logic for the North and South buttons on the OH1 handpaddle.

Syntax: `OH1.swapNS {off|on}`

`Off` – North is north, …

`On` – North is south, …

Example: Set the default mapping

`OH1.swapNS off`

OH2.Enable – Enable/Diables Observers Hand Paddle #2 inputs.

Syntax: `OH2.enable {off|on}`

`Off` – ignores the OH2 handpaddle IO

`On` – accepts OH1 handpaddle inputs

Example: Enable the OH2 handpaddle.

`OH2.enable on`

OH2.Rate – Set the maximum velocity for the OH2 handpaddle.

Syntax: `OH2.rate rate`

`rate` – This value represents arcseconds/seconds. Range is 0 to 60 as/s.

Example: Set the rate to 10 as/s.

`OH2.rate 10`

OH2.swapEW – Swap the logic for the East and West buttons on the OH2 handpaddle.

Syntax: `OH2.swapEW {off|on}`

Off – East is east, …
On – East is west, …
Example: Set the default mapping
`OH2.swapEW off`

OH2.swapNS – Swap the logic for the North and South buttons on the OH2 handpaddle.
Syntax: `OH2.swapNS` *{off|on}*
Off – North is north, …
On – North is south, …
Example: Set the default mapping
`OH2.swapNS off`

OS.2base – Transfer the enabled offsets and rates to the base and clear the offset values
Syntax: `os.2base` *offset*
offset – Specify offset(s) to be transferred (User, Beam, All)
Example: Transfer the enabled beam offsets and rates to the base
`Os.2base beam`

PID.Dec – Makes a request to the rtcs to change the PID value for the Dec axis.
Syntax: `pid.dec` *P I D*
P I D – Numeric PID values. Range is -100000 to 100000.
Example: Set P=25000, I=75000, D=5000
`pid.dec 25000 75000 5000`

PID.HA – Makes a request to the rtcs to change the PID value for the HA axis.
Syntax: `pid.HA` *P I D*
P I D – Numeric PID values. Range is -100000 to 100000.
Example: Set P=25000, I=75000, D=5000
`pid.HA 25000 75000 5000`

Polar.Motion – Specifies the earth's polar motion
Syntax: `polar.motion` *ha dec*
ha – HA polar motion in arcseconds (-10 to 10)
dec – Dec polar motion in arcseconds (-10 to 10)
Example: Set polar motion to 5.1 arcseconds in HA, -3.5 in Dec
`Polar.motion 5.1 -3.5`

Pressure – Specifies the atmospheric pressure
Syntax: `pressure` *value*
value – Pressure in mBars (200 to 2000)
Example: Set atmospheric pressure value to 547.8 mBars
`Pressure 547.8`

Pstart – Start or kill and restart child task
Syntax: `Pstart` *task*

`task` – Task name (e.g. pslow, audio, …)
Example: Start (or kill and restart) the audio child process
`Pstart audio`

Pstop – Kill a child application
Syntax: `pstop` *`task`*
`task` – Task name (e.g. pslow, audio, …)
Example: Kill the pslow child process
`Pstop pslow`

Pt.Add.Star – Append the pointing data for the current star data to the tpoint file
Syntax: `pt.add.star`
Example: Append the current star's pointing data
`Pt.add.star`

Pt.Caption – Specify the text of a caption for the pointing data file
Syntax: `pt.caption` *`text`*
`text` – Text of the caption (max 80 characters)
Example: Set the caption to "Pointing run 17 December 2004"
`Pt.caption Pointing run 17 December 2004`

Pt.Clear – Clears all the non-map pointing offset and rates. These are pt.adj, pt.spiral, and pt.rate.
Syntax: `Pt.clear`
Example: `Pt.Clear`

Pt.Convert – Converts current HA & Dec to numbers to enter into t3remote to test pt runs [**temporary test command**]
Syntax: `pt.convert`
Example: Print the current HA and Dec converted to t3remote values
`Pt.convert`

Pt.Dir – Define the directory path for pointing data files
Syntax: `pt.dir` *`path`*
`path` – Directory path [defaults to /home/tcs3/data/tpoint/$DATE]
Example: Set the directory path to /home/tcs3/temp/tpoint
`Pt.dir /home/tcs3/temp/tpoint`

Pt.Filename – Set the name for the pointing file
Syntax: `pt.filename` *`name`*
`name` – Name to use for the pointing file
Example: Set the pointing file name to pt_test_number_2
`Pt.filename pt_test_number_2`

Pt.Find – Find the nearest guide star to the given HA and Dec and load into the next object buffer
Syntax: `pt.find` *`ha dec`*
`ha` – HA position in hh`:mm:ss.ss`

dec – Dec position in dd:mm:ss.ss

Example: Find the guide star closest to the position -00:00:01.35 00:01:12.42

```
Pt.find -00:00:01.35  00:01:12.42
```

Pt.MAdj – Adds the corrections (Peak, Spiral, Rates) to the MAdj (MapAdjustment) registers. The MAdj values are not cleared after a slew, thus is used to adjust the Map's IH/ID values. The Peak, Spiral, and Rates registers are zerored.

Syntax: `pt.madj`

Example: `Pt.madj`

Pt.MAdj.Set – Set the MAdj (MapAdjustment) IH/ID pointing coefficient values. The MAdj values are not cleared after a slew, thus is used to adjust the Map's IH/ID values.

Syntax: `pt.madj.set` *ID IH*

IH – IH correction in arcseconds.

ID – ID correction in arcseconds.

Example: Set the adjustment IH /ID coefficient value to 5, 10 arcseconds

```
Pt.madj.set 5 10
```

Pt.Map – Turn the pointing map on or off

Syntax: `pt.map` *control*

control – Turn the pointing map ON or OFF

Example: Turn the pointing map on

```
Pt.map on
```

Pt.Map.Set – Set a pointing map coefficient value.

Syntax: `pt.map.set` *id value*

id – ID of coefficient (IH, ID, NP, CH, ME, MA, FO, HCES, HCEC, DCES, DCEC, TF, TX)

value – Value for the coefficient

Example: Set the ID pointing coefficient to 141.5

```
Pt.map.set ID 141.5
```

Also coefficients can be set using their ID as keyworks. The following are valid tcs3 commands:

```
IH          +3.4330
ID        -108.2731
NP          -2.5460
CH         +13.3261
ME         +10.4966
MA         +31.2848
HCES       -70.4861
DCES        +7.7726
FO         +31.0086
TF         +11.0809
TX         -39.3999
```

Pt.Next – Get the next (or specified) pointing object HA and Dec from table

Syntax: `pt.next` *`index`*
`index` – Index of object [optional – defaults to the next in the list]
Example: Get the HA and Dec of the object with index 5
`Pt.next 5`

Pt.Open – Open the pointing data file (see *pt.dir* and *pt.filename*)

Syntax: `pt.open`
Example: Open the pointing data file
`Pt.open`

Pt.Peak.Clear – Clear the Peak IH/ID pointing coefficient values

Syntax: `pt.peak.clear`
Example: Clear the adjustment ID and IH values.
`Pt.peak.clear`

Pt.Peak.Inc – Increment the value of the Peak IH/ID coefficients.

Syntax: `pt.peak.inc` *`ha dec`*
`ha` – Increment the adjustment IH coefficient in arcseconds.
`dec` – Increment the adjustment ID coefficient in arcseconds.
Example: Increment the user IH by 23.9 and decrement the user ID by 4.3
`Pt.peak.inc 23.9 -4.3`

Pt.Peak.Set – Set the Peak IH/ID pointing coefficient values.

Syntax: `pt.peak.set` *`coeff value`*
`coeff` – Coefficient to be set (ID or IH)
`value` – Value to set the coefficient to (in arcseconds)
Example: Set the adjustment IH coefficient value to 9.2 arcseconds
`Pt.peak.set ih 9.2`

Pt.Rate– Sets the pointing correction rate. Positive values will move the telescope East & South, so the rate should be set to the measured drift. For example, if drifting West by 0.02 and North by 0.01 as/s, used pt.rate 0.02 0.01

Syntax: `pt.rate` *`ih id`*
`ih` – The HA axis rate in AS/S. Ranges is +/- 5 as/s.
`id` – The dec axis rate in AS/S. Ranges is +/- 5 as/s
Example: Set the ID pointing rate to 0.001 as/s
`Pt.rate 0 0.001`

Pt.Rate.inc – Increments the pointing rate. Positive values will move the telescope East & South, so the rate should be set to the measured drift. For example, if drifting West by 0.02 and North by 0.01 as/s, used pt.rate 0.02 0.01

Syntax: `pt.rate.inc` *`ih id`*
`ih` – The HA axis rate in AS/S. Ranges is +/- 5 as/s.
`id` – The dec axis rate in AS/S. Ranges is +/- 5 as/s
Example: Increments the IH pointing rate by 0.001 as/s
`Pt.rate.inc 0.001 0`

Pt.Restore – Clears the IH ID values in the Corrc, (Peak, Spiral, Rates) registers. And read the pt.save.txt file (created by pt.save), to load the last saved IH ID values into the MAdj registers..

Syntax: `pt.restore`
Example: `pt.restore`

Pt.Save – Makes the map ID, IH equal to the total value (Map + Adj + Spiral + Rates). The adj, spiral, and rates variable are reset to zero. The new map ID, IH values are written to a file, pt.save.txt, in the IC directory. If the TCS3 is re-started, the IC will used the lasted saved IH,ID values for the map.

Syntax: `pt.save`
Example: `Pt.save`

Pt.Sep – Specify the separation between pointing targets

Syntax: `pt.sep` *separation*
separation – Separation in degrees between pointing objects (5.0 to 30.0)
Example: Set up a pointing target array with a 20.0 degree separation
`Pt.sep 20.0`

Pt.Spiral.Center – Zero's the pointing map spiral's offset, return you to the center position.

Syntax: `Spiral.center`
Example: `Spiral.center`

Pt.Spiral.In – This command move the pointing map spiral inwards.

Syntax: `Pt.spiral.in`
Example: Start moving inward
`Pt.spiral.in`

Pt.Spiral.out – This command move the pointing map spiral out wards..

Syntax: `Pt.spiral.out`
Example: Start moving outward
`Pt.spiral.out`

Pt.spiral.rate – Sets the pointing map spiral velocity in as/s.

Syntax: `Pt.spiral.rate` *vel*
Vel - velocity in as/s, ranges is 10 to 70.
Example: `Pt.spiral.rate 40.0`

Pt.Spiral.Stop – This command stop the pointing map spiral.

Syntax: `Pt.spiral.Stop`
Example: Stops the pointing spiral
`Pt.spiral.stop`

Pt.spiral.wid – Sets the pointing map spiral width in arcseconds. Should be matched to your field of view.

Syntax: `Pt.spiral.wid` *arcsec*
arcsec – spiral's width in arcseconds, ranges is 10 to 180.
Example: `Pt.spiral.width 60`

SafetyBrd.Reset – Sents a reset pulse to the T3 Servo Electronic (clears safety board latched errors).
Syntax: `SafetyBrd.Reset`
Example: Send the reset command
`safetybrd.reset`

Scan.Clear – Turns off scaning, and sets to 0 the scan offset and duration.
Syntax: `scan.clear`
Example: `Scan.clear`

Scan.Go – Move to specified offset from the current position
Syntax: `scan.go`
Example: Start a scan as set up by *Scan.Se*
`Scan.go`

Scan.Return – scan by returning to the base (offset 0,0) position.
Syntax: `scan.return`
Example: Return to the original position
`Scan.return`

Scan.Set – Set up the parameters for scanning to an offset position
Syntax: `scan.set` *`ra dec time`*
`ra` – RA offset to be applied to the original position (arcseconds)
`dec` – Dec offset to be applied to the original position (arcseconds)
`time` – Duration of the move from the original to the offset position (seconds)
Example: Set up to offset by 200.2 arcsec in RA, -30.4 in Dec in a scan of 15.5 seconds
`Scan.set 200.2 -30.4 15.5`

Shutter.Lower – Control the lower shutter
Syntax: `shutter.lower` *`control`*
`control` – Shutter action (stop | down | up)
Example: Raise the lower shutter
`Shutter.lower up`

Shutter.Upper – Control the upper shutter
Syntax: `shutter.upper` *`control`*
`control` – Shutter action (stop | down | up)
Example: Stop the upper shutter movement
`Shutter.upper stop`

Slew – Slew to the next object in the specified buffer
Syntax: `Slew` *`buffer`*
`buffer` – Object buffer (TO, OB, or SC) [optional – defaults to TO]
Example: Slew to the next object in the SC next buffer
`Slew sc`

Slew.Abort – Abort a slew and switch to track mode

Syntax: `slew.abort`

Example: Abort the slew in progress and enter the track mode

```
Slew.abort
```

Slew.Reslew – Re-issue the slew command to the pmac. Something when commanded to slew, the PMAC will not move the axes. This command re-issue the command.

Syntax: `slew.reslew`

Example: `Slew.reslew`

Sw.Limits.Override – Control software limits

Syntax: `sw.limits.override` *`control`*

`control` – OFF or On {ON ignores software limits)

Example: Ignore software limits

```
Sw.limits.override on
```

Sw.Limits.Set – Sets the software limits. TCS only allow stricter software limits from the defaults. This command is used to restrict the TCS range for instrument configurations outside the 'safety' zone.

Syntax: `sw.limits.set` *`EAST WEST SOUTH NORTH`*

`EAST` – East limit in hours. Ranage is -1:00:00 to -05:35:00

`West` – West limit in hours. Ranage is 1:00:00 to 05:35:00

`SOUTH` – South limit in degrees. Ranage is 10:00:00 to -55:00:00

`NORTH` – North limit in degrees. Ranage is 30:00:00 to 67:00:00

Example: This example sets new software limits:

```
Sw.limits.set -05:00:00 05:00:00 -45:00:00 45:00:00
```

This example sets the software limits back to their defaults:

```
Sw.limits.set -05:35:00 05:35:00 -55:00:00 67:00:00
```

System.power – Turns system power on or off.

Syntax: `system.power` *`{off|on}`*

Example: Turn system power on

```
System.power on
```

TempK – Set ambient temperature value in degrees Kelvin

Syntax: `tempk` *`t`*

`t` – temperature in deg Kelvin (100 – 350)

Example: Set ambient temperature value to 120.4 degrees Kelvin

```
Tempk 120.4
```

Track – Commands tcs to enter the 'track' servo mode

Syntax: `track`

Example: Command tcs to enter track mode

```
Track
```

Track.NoIOnOffset.HA

Track.NoIOnOffSet.Dec – Turn off/on a flag telling the tcs3 to hold the pmac's integrator value during an offset. This command is not intended for observers, operators, or daycrew. It use should be reserved to the tcs3 servo engineer.

Syntax: `track.NoIonOffset.HA { off | on  }`
`track.NoIonOffset.Dec { off | on  }`
Example: To Hold the integrator during offsets on the HA axis:
`Track.NoIOnOffset.HA on`

TOHP.Enable – Enable/Diables TO Hand Paddle inputs.

Syntax: `TOHP.enable` *`{off|on}`*
`Off –` ignores the handpaddle IO
`On` – accepts handpaddle inputs
Example: Enable the TOHP handpaddle.
`TOHP.enable on`

TOHP.Rate – Set the maximum velocity for the handpaddle.

Syntax: `TOHP.rate` *`rate`*
`rate –` This value represents arcseconds/seconds. Range is 0 to 60 as/s.
Example: Set the rate to 10 as/s.
`TOHP.rate 10`

User.Init– Initializes the users offset by: making the current position the base (0,0), setting the ra, dec values to 0,0, and enabling the offset.

Syntax: `user.init`
Example: Initialize the user offsets
`User.init`

User.Inc – Apply an increment to user RA and Dec offsets

Syntax: `user.inc` *`ra dec`*
`ra` – RA offset increment (arcseconds)
`dec` – Dec offset increment (arcseconds)
Example: Apply an incremental Dec offset of -25.1 arcseconds
`User.inc 0 -25.1`

User.Off – Disable user offsets

Syntax: `user.off`
Example: Disable user offsets
`User.off`

User.On – Enable user offsets

Syntax: `user.on`
Example: Enable user offsets
`User.on`

User.Set – Set the values for the user RA and Dec offsets

Syntax: `user.set` *`ra dec`*

ra – Value of user RA offset (arcseconds)
dec – Value of user Dec offset (arcseconds)

Example: Set user Ra offset to 110.7 and Dec to -29.0 arcseconds
`User.set 110.7 -29`

User.Spiral.center – Zero's the user spiral's offset, return you to the center position.

Syntax: `User.Spiral.center`
Example: `User.spiral.center`

User.Spiral.in – Command to move inwards along the spiral.

Syntax: `User.Spiral.In`
Example: `User.Spiral.In`

User.Spiral.Out – Command to move outwards along the spiral.

Syntax: `User.Spiral.Out`
Example: `User.Spiral.Out`

User.Spiral.rate – Sets the user spiral velocity in as/s.

Syntax: `User.Spiral.rate` *vel*
Vel – velocity in as/s, ranges is 10 to 40.
Example: `Spiral.rate 40.0`

User.Spiral.Stop – Stops moving along the spiral.

Syntax: `User.Spiral.Stop`
Example: `User.Spiral.Stop`

User.Spiral.wid – Sets the spiral width in arcseconds. Should be matched to your field of view.

Syntax: `User.Spiral.wid` *arcsec*
arcsec – spiral's width in arcseconds, ranges is 10 to 180.
Example: `User.Spiral.wid 60`

User.Toggle – Toggle the user offset enable setting

Syntax: `user.toggle`
Example: Toggle currently enabled user offsets to disabled
`User.toggle`

UT1Delta – Set the value of ut1 – utc. This value is need to calculate UT1 (ut1 = utc + ut1delta). You can obtain this value from the International Earth Rotation Service 's Bulletin B reports. (http://maia.usno.navy.mil)

Syntax: `ut1delta` *sec*
sec – Value of ut1 – utc in seconds (-0.51 to 0.51)
Example: Set ut1 – utc delta value to 0.35 seconds
`Ut1delta 0.35`

VTCS.Env.Update – Enable or disable fio_c updating of vtcs environment variables

Syntax: `VTCS.env.update` *mode*

`*Mode* can be:`

- *0* – Don't update
- *1* - update from fio_c (old tcs1 HM & temp sensors).
- *2* – update from fio_a (Vaisala HMT300 Sensors).

Example: Updates the vtcs using the HMT300 device

```
vtcs.env.update 2
```

Wavelength– Set the value of the observed wavelength

Syntax: `wavelength` *`m`*

m – Wavelength in microns (0.1 – 50.0)

Example: Set observed wavelength to 43.8 microns

```
Wavelength 43.8
```