Telescope Control System
External Interface Definitions

DCT-0720S-005-8

January 9, 2013

Author: Philip Taylor, Observatory Sciences Ltd.

Approved by: ______________________

Revision History

Revision Number	Author	Date	Description of Changes	ECN #
A	P. Taylor	18 July 2008	Initial Release	
B	P. Taylor	1 August 2008	Revised after TCS PDR.	
C	P. Taylor	4 December 2008	Release for TCS CDR 1. Command tables reformatted. Types now reflect actual LabVIEW data types used. 2. Guide target command definitions now included in the same table entries as the corresponding science target commands. 3. Command parameter ranges corrected or amended. 4. Extra TCS commands added: guide, rotCurrentWrap, azCurrentWrap, pointnew, pointadd, focusName 5. Separate Rotator command section added. 6. Many changes to subsystem command descriptions, particularly TCS, Mount, Rotator and AOS.	
D	P. Taylor	13 January 2009	1. Section 1.4 split into Controlling and Referenced Documents. 2. ICRS frame option removed from 'target' command: default now FK5. 3. All command names now entirely lower case. 4. All type names now Capitalized. 5. All instances of type 'float' changed to 'Double'. 6. Most instances of type 'integer' changed to 'Enum'. 7. Several instances of type 'string' changed to 'Enum'. 8. Some instances of type 'integer' changed to 'U32'. 9. 'guider' parameter to 'guide' command removed. 10. Extra focii added to 'focusname' command. 11. New airmass limit command (amlimit). 12. ICRS frame option removed from 'rotator' command: default now FK5. 13. Section 2.2.1: wording changed regarding ICRS (no longer supported). 14. Section 2.3.4: JPL Horizons implementation changed to use LabVIEW. 15. Section 2.5.4: modified description of MCU demand timing. 16. Section 2.5.5 added to describe MCU status handling in more detail. 17. Section 2.6: Added TCS/AOS command and status interface table. 18. Section 2.7: Added TCS/ADC command and status interface table. 19. Added guiderName item to GCS/OCS interface table.	
E	P. Taylor	26 January 2009	1. State that command parameter values are case-insensitive. 2. Added ‘m1cover’ and ‘setaccessmode’ commands. 3. Added explanatory text for ‘amlimit’, ‘setaccessmode’ and ‘m1cover’ commands. 4. Explanatory text for ‘guide’ command updated to reflect removal of guide source parameter. 5. Updated ‘tcsdebug’ command explanatory text. 6. Units now specified for all items in Status table. 7. Units for Double type RA/Dec position status values defined to be radians. 8. Added UNKNOWN to the set of possible State item values (the initial, transitory state on startup).	

			9. LST status value has only one decimal place of precision in seconds. 10. Added demandRAString and demandDecstring items to Status table. 11. Renamed Sun & Moon distance and proximity status items.	
F	P. Taylor	10 March 2009	1. Added WFS target related commands. (wtarget, wwavelength, wdifftrack, wplanet, worbit, woffset, wabsorboffset, wclearoffset). 2. Added comment re elevation range in Engineering Mode. 3. Difftrack parameter ranges modified to allow negative values. 4. Guider to TCS Command & Data table now has extra information re Commands. 5. Co-ordinate transformation data defined as a Guider to TCS data flow. Similar change also made in the TCS-AOS and TCS-ADC tables. 6. Two additional rotator angle values added to status table (IAA and IPA). 7. Units for status items defining angles have been changed from Radians to Degrees. 8. CurrentTimetoSet/Rise status items have been renamed.	
G	P. Taylor	1 April 2009	1. Removed filename parameter from 'pointnew' command. 2. Added 'name' parameter to 'orbit' commands 3. Removed optional 'file' as first parameter for Target commands. 4. Added new commands 'startup','shutdown' and 'ephemeris' 5. Added new status items: 'demandRotPA', 'targetName', 'm1cover' and 'accessmode'. 6. Updated section 3.7 Mount Control Interface to follow new MCC command structure being used to identify command destination. 7. Restructured TCS to Guider, AOS and ADC interface sections. The TCS commands & status have been removed from the Guider table and are now described in the next paragraph. 8. Status items now consistently defined with full names of units 'minutes' and 'degrees'.	
H	P. Taylor	21 April 2009	1. JIRA TCSD-138. Shared variable implementation, including SV names, now specified in this document. Several new and modified sections. 2. JIRA TCSD-77. Description of Startup and Shutdown added to sections 3.3.1 and 3.3.2. 3. JIRA TCSD-119 . Removed 'difftrack' command; added dtEpoch, dra and ddec parameters to 'target' command. Target command notes amended. 4. JIRA TCSD-131: Removed 'rotator' command; added rotPA and rotFrame parameters to 'target' command. Target command notes amended. 5. JIRA TCSD-130. 'orbit' command removed. 6. For clarity, renamed 'epoch' parameter of the 'target' command to 'pmEpoch' 7. The ranges for both parameters of the 'colloffset' command have been increased to +/-100.0. 8. Time Component status is currently internal to TCS, not external interface data. 9. Restructured Weather Server command and data table. Description of Weather Server commands and data updated. 10. JIRA TCSCDR-43. AOS introductory section reworded to remove mention of data filtering and specific data update rate. 11. Comment regarding making all TCS command parameters available as status items has been removed from the beginning of Section 2.4. 12. JIRA TCSD-110. Guider section revisions include using the term GCS rather than Guider where appropriate (including the section title). 13. JIRA TCSD-110. Added new WFS interface section. 14. JIRA TCSD-104. New section added (3.6.1) clarifying the use of the 'inPosition' flag 15. JIRA TCSD-104. Section 3.8.2, in the TCS/GCS section updated to clarify conditions for guiding.	

			16. JIRA TCSD-46. Updated ephemeris file definition. 17. JIRA TCSD-105. Section added (3.6.1) to clarify status updates rates. 18. Added guiderName item to TCS/GCS interface table. 19. Removed gdifftrack and wdifftrack commands from lists in 3.3.10 and 3.3.11 headers.	
I	P. Taylor	30 June 2009	1. Really did remove ‘rotator’ command from table. 2. TCSCommandResponse shared variable name changed to be TCSCommandResponseSV in section 2.3. 3. The minimum value for the ‘rotPA’ parameter of the ‘target’ command was changed from -360 to 0. 4. The minimum value for the ‘iaa’ parameter of the ‘rotoffset’ command was changed from -360 to 0. 5. The first enumerated value for the ‘rotframe’ parameter of the ‘target’ command changed from ‘track’ to ‘target’. This value is the default. 6. The minimum value for the ‘airmass’ parameter of the ‘amlimit’ command was changed from 0.0 to 1.0. Default value is now defined to be 3.0. 7. An additional enumerated value (stop) was defined for the parameter ‘state’ of the ‘m1cover’ command. 8. JIRA TCSD-144. Corrected units and added descriptions for ‘off1’ and ‘off2’ parameters to the ‘offset’ command. 9. Amended description for ‘poffset’ command parameters. 10. Removed mention of gdifftrack, gorbit, wdifftrack and worbit commands in section headers for 3.3.10 and 3.3.11. 11. JIRA TCSD-169. Rewrite of section 3.3.22, describing cable wrap ranges and the use of the ‘azCurrentWrap’ and ‘rotCurrentWrap’ commands. Added diagrams. 12. The ‘mccnamedpos’ command has been removed from the Mount Control Interface. This will now be implemented by sending the appropriate axis move commands to the MCU. 13. JIRA TCSD-142. Add units (mm) for the B1 & B2 vector elements in the GCS, WFS, AOS and ADC interface. 14. JIRA TCSD-143. Comments added regarding the two wavelengths used for ADC dispersion calculation. 15. Amended and added comments in section 3.8.2 regarding use of guide switch and implementation of TCS guiding corrections. 16. JIRA TCSD-154. ‘guiderName’ item added to GCS interface table. 17. Updated section 3.7.6 describing Mount Status: added details of SS & FB status requests. 18. Further updates to GCS interface description in sections 3.8.1 and]deleted section]. 19. JIRA TCSD-149. Added ‘guideclear’ command to command table and added section 3.3.19. 20. Change enumerated values for m1cover status in TCS status interface: now True or False. 21. JIRA TCSD-159. Add ‘gephemeris’ and ‘wephemeris’ commands. Changes to command table, sections 3.3.10 and 3.3.11. 22. JIRA TCSD-173. Changed first mode value for the ‘’accessmode’ command/status from ‘user’ to ‘operator’. 23. Removed Sun from the list of planet names accepted by the ‘planet’ command. 24. Added extra comments re GCS transformation matrix and TCS->GCS status items.	
J	P Taylor	14 Sept 2009	1. GCS Interface section (3.8) extensively modified, including extra interface items and rewrite of descriptive sections. 2. Removed transformation matrix items from GCS, WFS, AOS and ADC interfaces. 3. Data Interface tables for Guide, WFS and ADC split into separate Publish and Subscribe tables. 4. WFS Interface section (0) updates to descriptive sections. 5. JIRA TCSD-149. Added ‘guideautoclear’ command.	

			6. JIRA TCSD-52. Added ‘fracrate’ command. 7. Renamed SVs in Table in Section 2.3 to reflect component names. Added extra TCS[<Component>]CommandResponseSV’s. 8. JIRA TCSD-265. M1 cover status item in Table in Section 3.5 is now called m1CoverState with new set of enumerated values. Removed Boolean status M1CoverOpenIsTrueSV. 9. Section 3.11. Former ‘Time’ component now renamed to be IERS component. 10. Added NewScienceTarget item to TCS Status interface table. 11. Added command names in GCS, WFS, AOS and ADC interface tables. 12. Added “gtargTime” timestamp in TCS to GCS interface table at section 3.8.4.1 13. Updated AOS introductory section 3.9, including details of AOS units and co-ordinates. 14. Removed ‘fracrate’ and ‘wavelength’ commands 15. Added 3 new parameters to the ‘target’ command: configID, wavelength and fracrate. 16. Added 10 new parameters to the ‘ephemeris’ command: configID, frame, equinox, rotPa, rotFrame, dtEpoch, dra, ddec, wavelength and fracrate. 17. Added 8 new parameters to the ‘planet’ command: configID, rotPa, rotFrame, dtEpoch, dra, ddec, wavelength and fracrate. 18. Removed MCC commands ‘stowload’, ‘stowrecall’, ‘mccazelstow’ and ‘mccrotstow’. 19. Three uses of the word ‘class’ changed to ‘object’ in section 2.1. 20. Default frame for ‘ephemeris’ command changed to APPT (was FK5). 21. Section 3.3.4.1. Added description of how differential track rates are used with planet and ephemeris targets. 22. New and changed parameters described for the ‘ephemeris’ and ‘planet’ commands (sections 3.3.5 and 3.3.6). 23. Updated focus attributes set by the ‘focusname’ command (section 3.3.21). 24. Table in section 3.7.4.1 now lists the data items in the same order as the mount command. 25. Rewritten section 3.4.4 (including screenshot) describing the JPL Horizons interface. Now describes the system as currently implemented. 26. sTargetConfig, gTargetConfig and wTargetConfig commands added. 27. New tables added at 3.2.1 and 3.2.2 for definition and usage of target parameters. 28. New section 3.3.3 added and 3.3.4 re-arranged to reflect new target and target configuration parameters. 29. ADC corrections units are arcsecs. 30. Updated IERS component section 3.11, including leap seconds description. 31. Target parameter ‘dtepoch’ removed. Amended tables in sections 3.2.1 and 3.2.2.	

K	P Taylor	18 November 2009	*Revision K incorporates the changes numbered 1-12 below, previously shown in document Revision J Issues 2, 3 & 4.* 1. Add trackID and wtargTime WFS parameters in section 3.8.7.1. 2. Removed two Shared Variables: TCSActiveOpticsStatusSV and TCSWavefrontSensorCommandSV from table in section 2.3 3. In table 3.5 renamed “M1CoverOpenIsTrueSV” to be “M1CoverStateSV”. 4. In table 3.5 renamed “NewScienceTarget” to “NewScienceTargetSV”. 5. Removed Shared Variable TCSWavefrontSensorCommandResponseSV from table in section 2.3. 6. Added Shared Variable TCSWeatherServerCommandResponseSV to table in section 2.3. 7. Specify available planet names in footnote to table in section 3.2.2. 8. TCS/GCS and TCS/WFS command descriptions updated in Sections 3.8.5.2 and 3.8.8.1 to reflect new targetConfig command. 9. Added two wavelength items to table in section 3.10.3.1 to specify wavelengths for ADC dispersion calculation. 10. Range of ADC pointing offset values increased to be ±1000.0 arcsecs. in section 3.10.3.2. 11. Add ‘PA’ mode option for ‘offset’ commands in table in Section 3.2.3. 12. Wavelengths for ADC dispersion calculation now subscribed to as well as published by TCS. 13. Add qualifier to descriptions of rotPA and rotFrame parameters of TargetConfig: only applicable to Science target.	
L	P Taylor	22 December 2009	1. Changed two SV names: TCSIersComponentStatusSV has been renamed to be TCSIersStatusSV and TCSIersComponentCommandSV has been renamed to be TCSIersCommandSV. 2. In table 3.5, changed CurrentHA item units to be degrees. 3. In table 3.5, typo corrected: elError item units should be “arcsecs”. 4. In table 3.5, typo corrected: M1CoverStatesSV should be M1CoverStateSV. 5. Table reformatted in section 3.10.3.2: added line dividing the two commands in the table. 6. Section 3.12.1: ‘wsmode’ item type corrected to be Enum.	
M	P Taylor	11 January 2010	1. Added table in Section 3.3.26 describing TCS debug information.	
1	P Taylor	25 January 2010	1. Modified State status item definition to remove BUSY state.	263
2	P Taylor	8 April 2010	1. Corrected descriptions for rotator status items demandRotPA & currentRotPA in table in section 3.5. 2. JIRA TCSD-448. Add detailed description of the trackID status value in section 3.6.3. 3. JIRA TCSD-460. Added new SV status item currentParAngle (parallactic angle) in section 3.5. 4. JIRA TCSD-457 & 459. Added new status items to the	292

			NewScienceTargetSV shared variable (section 3.5): a. Demand Rotator On-sky Position Angle. b. Demand RA & Dec co-ordinates for the rotator axis. 5. JIRA TCSD-450. Corrected the descriptions of the rotator wrap limits in section 3.3.22, including adding a new diagram for the Cass rotator wrap. The Az and Cass wraps are not identical, as previously shown. 6. JIRA TCSD-451. Corrected SV names in section 2.3. The 3 SV names containing "Guide" have been changed to "Guider".	
3	P Taylor	26 April 2010	1. Updated set of state names for status SV TCSStateSV in section 3.5, changed in accordance with updated (Revision 4) DCT Software Architecture.	304
4	P Taylor	16 June 2010	1. JIRA TCSD-473. Changed the following items to be represented as a LabVIEW timestamp instead of MJD i. guideTargetPosition.gTargTime ii. guiderMeasuredError.guideCorrectionTime iii. wfsTargetPosition.wTargTime iv. TCSStatus.Time (was TCSStatus.MJD) v. IERS.timenls (was IERS.mjdnls) 2. JIRA TCSD-476. Changed the definition of the status item NewScienceTarget in section 3.5 to additionally incorporate the data in the class ScienceTargetConfiguration. The data item demandSkyPA was removed from NewScienceTarget.	339
5	P Taylor	24 June 2010	1. JIRA TCSD-477. Clarify definition and use of CommandID value in the TCS command protocol definition (new section 2.1.1.1). 2. Added GCS sub-section (3.8.5.3) mentioning GCS-related TCS commands (guide, guideclear & guideautoclear).	340
6	A Borrowman A Yoshimura	16 November 2010	1. JIRA TCSD-492. Update information on operation of TCS 'startup' command in section 3.3.1. Update sections 3.7.1.5, 3.7.2.2 and 3.7.3.3 detailing use of target-time-to-live when turning on tracking. 2. JIRA TCSD-496. Remove command mccazelstop from interface (changes in section 3.7). 3. In section 3.7.5 Tracking Demand Timing; remove TBC and add clarification of actual behavior. 4. JIRA TCSD-482. In section 3.5 TCS Status Interface, localTime, UTC and Date are removed. 5. JIRA TCSD-483. In section 3.5 TCS Status Interface, item NewScienceTarget. Added comment clarifying it is not included in the TCSTcsStatusSV.	377
7	P Taylor	25 April 2012	1. TCSD-529. In section 3.5 TCS Status Interface, add new item axesTrackMode. 2. TCSD-529. In section 3.5 TCS Status Interface, add new items InPositionAz, InPositionEl and inPositionRot. 3. TCSD-529. Added new section 3.6.1 describing the axesTrackMode. Changes to (renumbered) section 3.6.2 describing the inPosition status flags. 4. TCSD-534. Amended section 3.3.25 (pointadd command description) to mention rotator angle now written to pointing data file.	516

8.	P. Taylor	9 January 2013	1. TCSD-543. In section 3.5 TCS Status Interface, add new status items ExternalTargetCfgCmdPreviewIsTrue and ExternalTargetCfgCmdPreviewIsTrueSV. 2. TCSD-551. In section 3.5 TCS Status Interface, add new status item currentRotIPA and delete item current RotIPD. 3. TCSD-545. In section 3.5 TCS Status Interface, add 10 new individual limit-related status items: BlindSpotWarningSV, BlindSpotAlarmSV, ObjectSetWarningSV, ObjectSetAlarmSV, AzWrapWarningSV, AzWrapAlarmSV, RotWrapWarningSV, RotWrapAlarmSV, AirmassWarningSV, AirmassAlarmSV. 4. TCSD-545. In section 3.5 TCS Status Interface, change SunProximityTooCloseIsTrueSV and MoonProximityTooCloseIsTrueSV to have Event, not Alarm updates. 5. TCSD-552. In section 3.5 TCS Status Interface, add new item airmass.	528

Table of Contents

1 Introduction

The overall Control System for the DCT comprises a number of sub-systems. This document describes the interfaces between the DCT Telescope Control System (TCS) and external software systems.

1.1 Purpose

To define the interfaces between the TCS and DCT software systems with which it interacts.

1.2 Intended Audience

The intended audience for this document is

- The developers of other DCT control subsystems.
- Developers and maintainers of the TCS software.
- Reviewers involved in the DCT Software PDR and subsequent reviews.

1.3 Definitions, Acronyms and Abbreviations

This section gathers together some of the relevant definitions and abbreviations used in this document

ADC	Atmospheric Dispersion Corrector
AOS	Active Optics System
CAT	Catalogue Handling System
DCS	Dome Control System
DCT	Discovery Channel Telescope
ECS	Environmental Control System
FK5	Fifth Fundamental Catalogue reference system
GCS	Guider Control System
GDST	General Dynamics Satcom
ICS	Instrument Control System
ICRS	International Celestial Reference System
IERS	International Earth Rotation and reference system Service
M1	Primary Mirror
M2	Secondary Mirror
MCU	Mount Control Unit
MJD	Modified Julian Date
OCS	Observatory Control System
TAI	International Atomic Time

TBC	To be confirmed
TBD	To be decided
TCS	Telescope Control System
UR	User Requirement
URD	User Requirements Document
UT1	Universal Time
UTC	Coordinated Universal Time
WFS	Wave Front Sensor

1.4 Controlling Documents

[RD1] DCT User Requirements. DCT document DCT-0000S-005. (Generated from DOORS).

[RD2] DCT TCS User Requirements Document, DCT-0720S-007.

[RD3] DCT TCS Software Design Document, DCT-0720S-004.

1.5 Referenced Documents

[RD4] DCT Telescope Mount Specification, DCT-0410S-001-2, 22 January 2008.

[RD5] DCT Active Optics System Specification, DCT-0360S-003-5, 22 May 2008.

[RD6] DCT Guider and Wavefront Sensing System: Functional, Performance and Control Software Requirements, 25 March 2008.

[RD7] DCT-MCU Software Interface Specification Document DCT-0440S-011, GDST (Vertex/RSI) Document. 08-568-000 [Proprietary document].

[RD8] DCT TCS Data Definitions (DCT-0720D-010-014).

1.6 Overview

The remainder of the document is divided into sections,

- The first section describes the commands which can be sent to the TCS from any external system.
- Each subsequent section corresponds to a specific DCT subsystem with which the TCS interacts: either receiving data from or sending data to, the subsystem.

2 LabVIEW Shared Variable Implementation

The communication of commands and status between the TCS and the DCT subsystems with which it interacts is implemented using LabVIEW Shared Variables. The following sections provide details of the techniques used to pass commands and status data between the DCT subsystems, including the names of the shared variables being used.

2.1 Command Interface

The OCS to TCS command interface uses a LabVIEW implementation of the Command pattern. Commands are sent from the OCS by writing a suitable object (TCSCommand) as a flattened String to a LabVIEW shared variable.

To send a command to the TCS, the OCS creates a command class object and sets the parameter values to its class data and a unique Command ID is also set (see below). This data is then flattened to a string and written to the shared variable called `TCSCommandSV`.

When the TCS receives a new string in this shared variable, it unflattens the string and then retrieves a TCSCommand object. The TCS retrieves data values from the object and internally submits the command.

2.1.1 Command Protocol Data

The TCSCommand class contains command protocol data fields: a CommandID and an error cluster (defined in the following table).

Data Item	Usage	Type	Description/Notes
CommandID	Written by OCS before sending a command	U32	Unique ID for this command instance. For further details, see section 2.1.1.1 below.
Error	Written by TCS after validating the command	Standard LabVIEW Error Cluster = Status (Boolean) Code (I32) Source (String)	Error returned after TCS command validation

The OCS produces a unique ID (using the current time) for each command instance and sets it in the CommandID field.

After validation, the TCS writes a command response to the TCSCommand class error cluster. The TCS then flattens the response to a string and writes it to the shared variable called `TCS[<Component>]CommandResponseSV`. The OCS will determine the corresponding response to a particular command by matching up the CommandID.

2.1.1.1 CommandID Value

The CommandID sent to the TCS should be a U32 identifier, defined such that no more than one active command has the same CommandID. The TCS will ignore a command that has the same CommandID as the previous command, presuming it to be a duplicate reception.

One possible (and suggested) approach to generating a CommandID would be to generate a number based on time in units of 100us, counted from midnight UTC. Any consistent approach that meets the uniqueness criteria will satisfy the interface specification for the TCS.

2.2 Status Interface

The TCS periodically (1Hz) writes all its public status values into a LabVIEW shared variable called TCSTcsStatusSV which contains multiple status items. In addition, some of this status is also written into individual shared variables. A status item is separated out into an individual SV when another subsystem monitors that individual item for changes or alarm status. The name of the shared variable used for each status item is shown in the TCS Status table below in section 3.5.

The data is transferred between the TCS and OCS by flattening to a string representing the class, as described above in section 2.1.

The TCS also publishes status data to other shared variables that contain status specific to a particulr subsystem.

2.3 Subsystem Interface Shared Variables

The following tables shows the names of all shared variables used to implement the various TCS to DCT subsystem command and status interfaces.

TCS Publishes	**TCS Subscribes**
TCSTcsStatusSV *and additional low-level status items, see section 3.5.*	TCSTcsCommandSV
TCSTcsCommandResponseSV	TCSMccCommandSV
TCSMccStatusSV	TCSGuiderCommandSV
TCSMccCommandResponseSV	TCSActiveOpticsCommandSV
TCSGuiderStatusSV	TCSAtmosphericDispersionCommandSV
TCSGuiderCommandResponseSV	TCSIersCommandSV
TCSWavefrontSensorStatusSV	TCSWeatherServerCommandSV
TCSActiveOpticsCommandResponseSV	
TCSAtmosphericDispersionStatusSV	
TCSAtmosphericDispersionCommandResponseSV	
TCSIersStatusSV	
TCSIersCommandResponseSV	
TCSWeatherServerCommandResponseSV	

3 Telescope Control System Interface Definitions

3.1 TCS Command Interface

The table below lists the commands that the TCS will respond to. These commands will originate from either an external client or from operator interaction with a TCS engineering screen.

Parameter types are shown corresponding to the LabVIEW types used. All String or Enum type parameter values are case-insensitive.

3.2 TCS Command Definitions

Command	Parameter	Type	Format/Units	Valid Set/Range (default)	Description/Notes
startup					Start up the TCS
shutdown					Shutdown the TCS
sTargetConfig gTargetConfig wTargetConfig	targetConfigName	String			Name of this target configuration
	wl	Double	Microns	0.3 – 10.0 (0.6)	Effective target wavelength
	rotPA	Double	Degrees	0 - +360	Desired instrument position angle, Eastwards from North (Science Target only)
	rotFrame	Enum		target | fixed (target)	Rotator co-ordinate frame. target = same frame as target. fixed = maintain sky orientation (= AZEL) (Science Target only)
	fracrate	Double		0.0 – 1.0 (1.0)	Apply a fraction of the differential tracking (Science Target only)
	target	*This item contains all the target parameters and is defined in a separate table, see section 3.2.1 below*			

3.2.1 Target parameters

The complete set of target parameters is defined in the following table. Not every parameter value is used with all targets and types of target. The details of which target parameter values are significant for each target and target type are tabulated in the table below in section 3.2.2.

Parameter	Type	Format/Units	Valid Set/Range (default)	Description/Notes
targetName	String			Name of the object
ra	Double Double Double	HH MM SS	0 – 24	Minutes and Seconds optional. An optional fractional part can be provided with any of the Hrs/Degs, Minutes or Seconds fields.
dec	Double Double Double	DD MM SS	-90 – 90	
az	Double Double Double	DD MM SS	0 – 359.999999	Only used when frame = AZEL Note: Elevation range is restricted in Operator Mode.Larger elevation range is available when in Engineering Mode.
el	Double Double Double	DD MM SS	5 – 90	
frame	Enum		FK4 | FK5 | APPT | GAPPT | AZEL (Science Default = FK5) (Planet & Ephemeris Default = APPT)	
equinox	Enum Double	Optional prefix (B or J), followed by decimal year	[B | J] 1850.0 – 2100.0 (Default: J2000)	If initial letter is omitted, B is assumed before 1984.0, J otherwise.
pmEpoch (years)	Double	Decimal year	1850.0 – 2100.0 (Default: now)	Epoch in years, used if Proper Motion is defined
parallax	Double	Arcsecs	0.0 – 10.0 (0.0)	
pmra	Double	Secs/year	-10.0 – 10.0 (0.0)	Proper motion (RA)
pmdec	Double	Arcsecs/year	-100.0 – 100.0 (0.0)	Proper motion (Dec)
rv	Double	Km/second	-1000.0 – 1000.0 (0.0)	Radial velocity
dra	Double	Secs/second	-10.0 – 10.0 (0.0)	Differentail Track Rate (RA or Azimuth)
ddec	Double	Arcsecs/second	-10.0 – 10.0 (0.0)	Differential Track Rate (Dec or Elevation)

3.2.2 Target Parameter Usage

The following table shows the parameters available with the three forms of the TargetConfig command: sTargetConfig for the Science target, gTargetConfig for the Guide target and wTargetConfig for the WFS target. Each row corresponds to a parameter, with the name shown in the left hand column. The three main table columns tabulate the parameters used with the Science, Guide and WFS targets. Each of the columns is split into the three types of target: Sidereal, Planet and Ephemeris.

A table entry of "U" (Used) means that the supplied value for the parameter will be used with that target and type of target, otherwise a default value applies. A table entry of "-" means that the parameter value is ignored in that case.

Target / **Type** / **Parameter**	**Science**			**Guide**			**WFS**		
	Sidereal	*Planet*	*Ephemeris*	*Sidereal*	*Planet*	*Ephemeris*	*Sidereal*	*Planet*	*Ephemeris*
targetName	U	U (fixed menu, see 1. below)	U (ephemeris file path)	U	U (fixed menu)	U (ephemeris file path)	U	U (fixed menu)	U (ephemeris file path)
ra	U	-	-	U	-	-	U	-	-
dec	U	-	-	U	-	-	U	-	-
az	U	-	-	U	-	-	U	-	-
el	U	-	-	U	-	-	U	-	-
frame	U	- (always topocentric apparent)	- (supplied in data file)	U	- (always topocentric apparent)	- (supplied in data file)	U	- (always topocentric apparent)	- (supplied in data file)
equinox *(only used with FK5 & FK4 frames)*	U	- (always of date)	- (derived from frame)	U	- (always of date)	- (derived from frame)	U	- (always of date)	- (derived from frame)
pmEpoch	U	-	-	U	-	-	U	-	-
parallax	U	-	-	U	-	-	U	-	-
pmra	U	-	-	U	-	-	U	-	-
pmdec	U	-	-	U	-	-	U	-	-
rv	U	-	-	U	-	-	U	-	-
dra (daz)	U	U	U	U	U	U	U	U	U
ddec (del)	U	U	U	U	U	U	U	U	U

1. Planet target name is one of the following: Mercury, Venus, Moon, Mars, Jupiter, Saturn, Uranus, Neptune or Pluto.

3.2.3 Other TCS Commands

Command	Parameter	Type	Format/Units	Valid Set/Range (default)	Description/Notes
offset goffset woffset	type	Enum		SIMPLE | TPLANE | PA (SIMPLE)	
	off1	Double	Secs (RA)	-100.0 – +100.0 (0.0)	Offset in RA, relative to target position
	off2	Double	Arcsecs (Dec)	-100.0 – +100.0 (0.0)	Offset in Dec, relative to target position
	num	Enum		0 – 1 (0)	0 = user 1 = handset
absorboffset gabsorboffset wabsorboffset	num	Enum		0 – 2 (0)	0 = user 1 = handset 2 = both
clearoffset gclearoffset wclearoffset	num	Enum		0 – 2 (0)	0 = user 1 = handset 2 = both
porigin	x	Double	Millimetres	-100.0 – +100.0 (0.0)	Millimetres in the focal plane. Origin is the axis of the Cass rotator
	y	Double	Millimetres	-100.0 – +100.0 (0.0)	
poffset	dx	Double	Millimetres	-10.0 – +10.0 (0.0)	X and Y pointing origin offset values relative to current pointing origin
	dy	Double	Millimetres	-10.0 – +10.0 (0.0)	
	num	Enum		0 – 1 (0)	0 = user 1 = handset
absorbporigin	num	Enum		0 – 2 (0)	0 = user 1 = handset 2 = both
clearporigin	num	Enum		0 – 2 (0)	0 = user 1 = handset 2 = both
colloffset	ca	Double	Arcsecs	-100.0 – +100.0 (0.0)	Set collimation CA,CE values
	ce	Double	Arcsecs	-100.0 – +100.0 (0.0)	

clearcolloffset					Clear collimation values
guide	state	Enum		On | Off (Default Off)	Switch guiding on/off
guideclear					Set current guiding corrections to zero
guideautoclear	state	Enum		On | Off (Default On)	Switch guide corrections auto clearing On or Off.
focusname	focusname	Enum		Cass | Prime | Nasmyth1 | Nasmyth2 | BentCass1 | BentCass2 | BentCass3 | BentCass4 | BentCass5 | BentCass6 (Default Cass)	Define which telescope focus is to be used
azcurrentwrap	wrapnum	Enum		-1 | 0 | +1 (Default 0)	Define which cable wrap is to be used (azimuth travel range)
rotcurrentwrap	wrapnum	Enum		-1 | 0 | +1 (Default 0)	Define which rotator wrap is to be used (rotator travel range)
amlimit	airmass	Double		1.0 – 15.0 (Default 3.0)	Set airmass limit
pointnew					Create and open a new pointing data file. A unique name for the file will be defined internally.
pointadd					Add a new data point to the pointing data file
tcsdebug	level	Enum		0 – 10 (Default 0)	Set the TCS debug level. Increasing the debug level increases the number of debug messages produced.

setaccessmode	mode (password)	Enum (string)		operator | engineer	Set access mode. Password required only if specified mode is 'engineer'.
m1cover	state	Enum		open | close | stop	Open, close or stop the movement of the primary mirror cover
rotoffset	iaa	Double	Degrees	0 - +360	Instrument alignment angle

3.3 Notes on TCS Commands

3.3.1 Startup

The startup command will start up the TCS and bring it to a state where commands will be accepted and it is ready to begin observing. On receipt of a startup command from OCS the TCS will forward a startup command to all TCS sub-systems (e.g. MCC), taking all components from state 'standby' to 'running'.

On receipt of the startup command the MCC will send to the MCU the CTAKE command to take computer control of mount, i.e. action carried out by mccenable command. Once the mccenable command has successfully completed the mccstop command is sent to ensure the all mount axes are stopped.

3.3.2 Shutdown

The shutdown command will take the TCS to a state where commands will not be accepted and will no longer be able to observe.

On receipt of a 'shutdown' command the TCS will forward this to all TCS sub-systems, taking components from state 'running' to state 'standby'.

On receipt of the shutdown command the MCC will send a command to release computer control of mount to MCU (CREL), i.e. currently carried out using command 'mccdisable'.

Following completion of shutdown command the MCC will continue to monitor mount status by sending status commands to MCU but will no longer be able to move the mount.

3.3.3 Target Configuration

A *Target configuration* consists of a target configuration name (targetConfigName) and parameters that are associated with a particular configuration: rotator parameters (rotPa and rotFrame), wavelength and fractional rate. It also incorporates the target definition (see section 3.3.4 below).

3.3.3.1 Target Configuration: rotator parameters

The rotator parameters are rotPA and rotFrame. Three angles are relevant when defining a rotator position, defined as follows:

1. The Instrument Principal Direction (IPD). This is a direction on the instrument that the user decides is a convenient reference. Typically this would be the pixel +*y* direction for a CCD camera or the orientation of a spectrograph slit.
2. The Instrument Alignment Angle (IAA). This is the mechanically calibrated value which represents the orientation of the IPD with respect to the +*y* axis of the instrument rotator. Its value is defined using the ‘rotoffset’ command (see 3.3.29 below).
3. The Instrument Position Angle (IPA) is the position angle on the sky to which the the IPD will be set for the current observation.

The zero point is such that with the instrument position angle (IPA) set to zero in a celestial frame the projection on the sky of the nominated instrument principal direction (IPD) points towards the North pole and in topocentric az/el it points upwards. In this latter case, with the IAA set to 180 degrees and the IPA set to zero the mechanical rotator position will be close to zero. The projection on the sky of the instrument mount x/y axes rotates anti-clockwise as the position angle increases.

Two parameters are available with the target command to define the required rotator configuration to be used with this target:

1. rotPA. The required rotator position Angle (IPA).
2. rotFrame. This is the rotator co-ordinate frame. By default, this takes the value 'Track' which means that the target frame of reference is used. Setting the tracking frame to a celestial coordinate system (FK5, FK4 or Apparent) keeps the field stationary on the instrument. The option "Fixed" for the tracking frame will ignore any IAA that has been set, add 180 degrees to the demand IPA and set the frame to topocentric Az/El. The effect is to drive the mechanical rotator position close to the demanded IPA.

3.3.3.2 Target Configuration: wavelength parameter

The wavelength parameter is provided to set an effective wavelength for the current target configuration. This will be used by the pointing kernel when computing refraction corrections. The effective wavelength is a function of both the target's spectral energy distribution and any filters through which it is being observed.

3.3.3.3 Target Configuration: fracrate parameter

This parameter is used to apply a fractional differential tracking rate to the current Science target. The value can be in the range 0.0 – 1.0. The current differential track rate for the Science target is multiplied by the supplied fractional value.

The differential track rate for the target may have been specified directly using parameters dra/ddec with the Target command, or indirectly by selecting a Planet or Ephemeris target.

3.3.4 Target parameters

Target parameters provide all the necessary information to specify an astronomical target. See sections 3.2.1 and 3.2.2 above for tables defining the command parameters and how they are used with different targets and types of target.

Although most of the frames of reference supported are astronomical, the topocentric azimuth and elevation frame is also implemented.

Note that the ICRS is not supported as a reference frame: FK5 should be used instead.

Validation of the target parameters for this command also involves checks that the resultant mount demands are within limits and that the target is above the lower elevation limit. Additional checks are also performed for proximity to sun, moon or zenith.

If a target command is validated then the supplied parameters, plus any defaulted values, will be converted to the units required by the pointing kernel and then committed. As soon as this happens the kernel will start generating the demands appropriate to the new target. What the mount component will see will be a step change jump in demands within about 50 ms. Whether the mount acts on these demands will depend if it is tracking: if it is, the mount component and servo controllers attempt to drive the mount hardware so as to match the current position with the new demand.

Further details of target parameter formats and defaulting rules are as follows:

Parameter	Description
name	A name to identify the target object e.g. 3C273. The name can include spaces and need not be surrounded by quotes unless the name includes an "=". If no name is specified then the string "Unspecified" will be provided.

frame	The reference frame in which the coordinates of the target are specified. The default if none is specified is FK5 unless it can be deduced from the equinox (see below)
RA/Az	The interpretation of the target position parameters is treated as an RA or an Azimuth, depending on the frame. The frame must be AZEL for Azimuth positions, otherwise they are treated as RA/Dec. The format of the position can be specified by 1. All three parts supplied in sexagesimal format : Hours/Degrees (Integer) + Minutes (Integer) + Seconds (Double). 2. Hours/Degrees (Integer) + Minutes (Double), or 3. Hours/Degrees only (Double) So, for example, all the following are valid: RA = 3 4 5.61 RA = 3.5 RA = 3 4.5 Az = 231 40 21.6 Az= 231.1
Dec/El	Again the parameter names dec and el are synonyms and it is the frame parameter that determines how the following data is interpreted. Identical format to RA/Az.
equinox	The epoch of the mean equator and equinox. The defaults depend on the frame specified. FK5 – J2000.0 FK4 – B1950.0 APPT/AZEL – ignored If no letter (B or J) precedes the number then pre 1984.0 is assumed to be B whilst post 1984.0 is assumed to be J
pmEpoch	The zero point for the proper motion expressed as a year. If none is specified it will default to the epoch of the equinox.
parallax	Stellar parallax in arcsecs it defaults to 0.0
pmra	Proper motion in RA (s/yr) it defaults to 0.0
pmdec	Proper motion in Dec. (arcsec/yr). It defaults to 0.0
rv	Radial velocity (km/s). It defaults to 0.0
rotPA	Rotator position angle to be used with this target (see section 3.3.3.1 above)
rotFrame	Rotator co-ordinate frame to be used with this target (see section 3.3.3.1 above)
dra	Differential track rate for RA or Azimuth (see section 3.3.4.1 below)
ddec	Differential track rate for Dec or Elevation (see section 3.3.4.1 below)

3.3.4.1 Target differential tracking parameters

Any target can be given a differential track rate. The frame of the differential tracking is assumed to be the same as that of the target. It is assumed that the differential track rates are applied starting at the current time.

The `dra` and `ddec` parameters specify the differential tracking rate in Ra and Dec for the target. If the target is specified in the AZEL frame, these items are interpreted as azimuth and elevation tracking rates. Thereafter the TCS continually computes a new demand into the kernel such that

$$p(t) = p(t_0) + (t - t_0) * \dot{p}$$

for each coordinate.

Note that this assumes that the differential track rate is constant. For more precise tracking control, using variable differential tracking rates, the 'ephemeris' command should be used (see 3.3.5 below).

If a target is specified using the 'planet' or 'ephemeris' target, the TCS internally calculates the differential tracking rate of the object derived from the ephemeris data: this tracking rate is an inherent attribute of the object and cannot be specified or altered by the user. If differential tracking rate parameters are supplied with the 'planet' and 'ephemeris' commands, the supplied tracking rate is *added* to the inherent tracking rate of the object.

3.3.5 Ephemeris targets

The targetName parameter used with 'ephemeris' type targets is the filepath of the file containing the ephemeris data. The file extension '.eph' is assumed if not supplied.

This command defines the TCS target position as a set of specified RA & Dec values at specified times. The ephemeris data can be used with all targets (Science, Guide and WFS).

3.3.5.1 Ephemeris data

1. The first line of the ephemeris file contains the name of the object and the frame of reference in which the positions are specified. The frames of reference supported are the same as those supported with the target command. The target name must be provided; if missing, the frame is assumed to be J2000.0.

2. The ephemeris data follows, consisting of a series of lines defining a time along with the object's position at that time, as follows:

   `YYYY MM DD HH MM SS RA DEC`

   Where '`YYYY MM DD HH MM SS`' is the specified date and time (UTC) for this position and

   '`RA DEC`' are the RA and Dec for the object at the specified time. The format of the RA and Dec position values are of the form

   `hh mm ss.sss [+/-]dd mm ss.sss`

   All fields must be provided on each line, although the fractional seconds (.sss) parts of the RA and Dec values are optional.

   The JPL Horizons interface, provided as part of the TCS (see section 3.4.4), produces files which can be used with this command.

3.3.6 Planet targets

The 'planet' command provides a quick way of pointing the telescope at any of the major planets or the Moon. The first (targetName) parameter is the name of the planet and the remaining parameters correspond to the parameters with the same names used in the 'target' command. See section 3.4.1 below for further details.

3.3.7 offset

The offset command applies offsets to the target position but keeps track of them separately from the target position itself. The advantage of this is that by simply clearing the offsets one returns immediately back to the base pointing position. This is very convenient for performing repeated offsets, such as a grid of observations around a central position.

Offsets can be provided as simple offsets or as tangent plane offsets. Simple offsets just add to the current target position, tangent plane offsets are first converted into simple offsets.

Note that offsets are further distinguished by a number 0 or 1. The user offset is offset number 0, the software handset interface offset is number 1. The reason for having two offsets is that the software handset may have been used to tweak a target position to get a poorly specified target onto the pointing origin. If one then wants to do a grid about this position then user offsets don't affect these tweaks and when the user offsets are cleared the target returns to the pointing origin.

Internally the TCS handles both types of offset in the same way and simply adds them to the base target position before passing them to the pointing kernel. If p_0 is the base target position, off_0 the user offset and off_1 the total handset offset, then the resulting target position p is defined as

$$p = p_0 + off_0 + off_1$$

The offset command simply sets off_0 (or off_1) to whatever value you specify. It is therefore an absolute offset. Issuing the offset command ten times with an offset of 1 arcsec will yield a total offset off_0 of 1 arcsec. The software handset interface, however, keeps a cumulative total of offset values and ensures that if you click the offset button ten times you end up with off_1 equal to 10 arcsecs.

3.3.8 absorboffset

The absorboffset command resets p to be the base position plus whatever offsets are being absorbed whilst simultaneously setting those offsets to zero.

The offset number specified which offset is to be absorbed. The user offset is offset number 0, the software handset interface is number 1. Both user and handset offsets may be simultaneously absorbed by specifying the offset number as 2.

3.3.9 clearoffset

This command sets the specified offset(s) back to zero.

The offset number specified which offset is to be cleared. The user offset is offset number 0, the software handset interface is number 1. Both user and handset offsets may be simultaneously cleared by specifying the offset number as 2.

3.3.10 gtarget, gwavelength, gplanet, gephemeris, goffset, gabsorboffset, gclearoffset

These commands have parameters and functions identical to the corresponding science target commands described above. These commands define the Guider target attributes, including differential tracking, using an ephemeris file to define the target and offsets.

3.3.11 wtarget, wwavelength, wplanet, wephemeris, woffset, wabsorboffset, wclearoffset

These commands have parameters and functions identical to the corresponding science target commands described above. These commands define the WFS target attributes, including differential tracking, using an ephemeris file to define the target and offsets.

3.3.12 porigin

Porigin command defines the pointing origin. This is the term used for the point in the focal plane at which the target is imaged. The normal default pointing origin will be 0, 0. Changing the pointing origin will image the target to a different position in the focal plane without changing the RA and Dec of the target.

3.3.13 poffset

The pointing origin offset command just adds an adjustment to the base pointing origin. The offset pointing origin number specified which offset is to be defined. The user pointing origin offset is offset number 0, the software handset pointing origin interface is number 1.

3.3.14 absorbporigin

The absorb pointing origin command absorbs the offsets into the base pointing origin at the same time as setting those offsets to zero.`

The offset number specified which pointing origin offset is to be absorbed. The user pointing origin offset is offset number 0, the software handset interface pointing origin offset is number 1. Both user and handset pointing origin offsets may be simultaneously absorbed by specifying the offset number as 2.

3.3.15 clearporigin

The clear pointing origin command sets the appropriate pointing origin offsets back to zero.

The offset number specified which pointing origin offset is to be cleared. The user pointing origin offset is offset number 0, the software handset interface pointing origin offset is number 1. Both user and handset pointing origin offsets may be simultaneously cleared by specifying the offset number as 2.

3.3.16 colloffset

If on slewing to a target the target does not appear on the fiducial there can be a number of reasons. It may be that the pointing model is not accurate. In this case one can adjust the collimation corrections so that the target appears at the required position.

The colloffset command sets the values of the horizontal and vertical collimation corrections (usually referred to as CA and CE) in the telescope pointing model. The units are seconds of arc.

3.3.17 clearcolloffset

The clearcolloffset command sets the horizontal and vertical collimation corrections (CA and CE) in the telescope pointing model to zero.

3.3.18 guide

This command will be used to switch guiding on or off. When guiding is on, adjustments will be applied to the TCS pointing using correction data sent from the guider. When guiding is off, this data will be ignored.

3.3.19 guideclear

This command sets the guider offset values to zero.

3.3.20 guideautoclear

This command controls automatic clearing of the guider offset values. If automatic guide clearing is switched On, the guider offset values will be automatically set to zero whenever a new target is selected (but not when a new offset is defined).

If automatic guide clearing is switched Off, the guider offset values will only be cleared when the ‘guideclear’ command is issued.

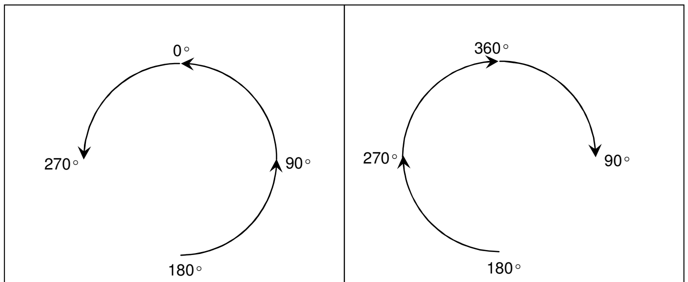
3.3.21 focusName

This command will be used to specify the name of the focus to be used. The command will set the correct attributes in the TCS to use with the specified focus. Currently the focus attributes that are set by this command are:

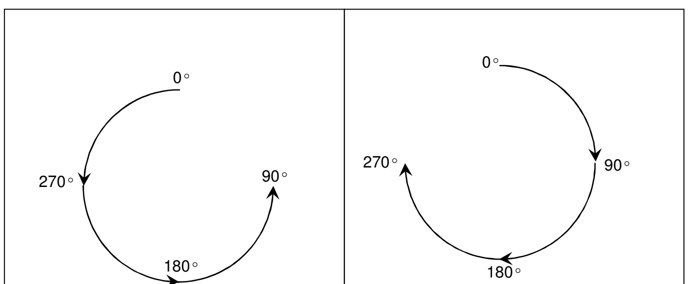
1. Pointing origin. This is specified in mm relative to the centre of the rotator for the focus.
2. Effective focal length for the focus. Specified in mm. This is used to convert angular to linear distances on the focal plane and vica versa.
3. Rotator type for the focus. This is used by the pointing kernel when calculating rotator demands. It can be one of: OTA (for Prime, Newtonian, Cassegrain etc.), NASYMTH_L, NASYMTH_R, COUDE_L or COUDE_R.

3.3.22 azCurrentWrap, rotCurrentWrap

The commands azCurrentWrap and rotCurrentWrap allow the wrap to be specified if required. These commands will typically be used when a cable wrap is currently approaching its limit.



Negative (W) Azimuth Cable Wrap | Positive (C) Azimuth Cable Wrap



Negative (W) Cassegrain Cable Wrap | Positive (C) Cassegrain Cable Wrap

The azimuth and rotator axes are capable of 540 degrees of movement, covering the range ±270 degrees. The Azimuth wrap is centered on the 180 degrees (South) position, with the Cassegrain wrap centered on the 0 degrees (North) position. This is illustrated in the above diagrams.

The cable wraps are defined as follows for the azimuth wrap:

1. Wrap state -1, covers the range from 180 degrees, with decreasing azimuth values through the 0 degree position, to 270 degrees. This is called ‘W’ or –ve wrap in the GDST Mount Interface document.

2. Wrap state +1, covers the range from 180 degrees, with increasing azimuth values through the 360 degree position, to 90 degrees. This is called ‘C’ or +ve wrap in the GDST Mount Interface document.

The cable wraps are defined as follows for the Cassegrain wrap:

1. Wrap state -1, covers the range from 0 (= 360) degrees, with decreasing azimuth values through the 180 degree position, to 90 degrees. This is called ‘W’ or –ve wrap in the GDST Mount Interface document.

2. Wrap state +1, covers the range from 0 degrees, with increasing azimuth values through the 180 degree position, to 270 degrees. This is called ‘C’ or +ve wrap in the GDST Mount Interface document.

“Wrap state 0” denotes that the shortest route should be taken by the mount, with no regard to the wrap. This is referred to as ‘S ‘in the GDST Mount Interface document. This is effectively the “do nothing” parameter value and is sent by default with position demands to the MCU.

So the MCC will normally send the value ‘S’ (shortest) to the MCU as the wrap parameter with its 20 Hz position (PD) demands. When a user wishes to change wrap, the azCurrentWrap or rotCurrentWrap commands will cause the wrap parameter value sent with the PD command to be ‘C’ or ‘W’. This will cause the mount to move to the current azimuth value in the other wrap, thus swinging the mount through 360 degrees of azimuth.

Once the telescope slew and the requested change of wrap have been completed, the MCC will resume sending the value ‘S’ (shortest) with its PD commands at 20 Hz, regardless of which cable wrap is currently in use.

The telescope mount continuously reports the current cable wrap as either C or W. During tracking, the wrap changes automatically where possible: for example, when using the negative cable wrap, the wrap will switch to the positive wrap (without any user action) as the telescope tracks through the 180 degree azimuth point

Note that it will not be possible to move to a specified wrap if the current azimuth is outside the required wrap’s azimuth range.

3.3.23 amlimit

This command defines a maximum air-mass value for the current target. This value will be used when calculating the time to airmass limit status value.

3.3.24 pointnew

This command creates a new pointing test data file: the filename is provided as the parameter.

3.3.25 pointadd

This command writes a pointing test data record to the file previously specified by the pointnew command.

The data written to the file consists of the demanded Azimuth and Elevation, followed by the actual mount Azimuth and Elevation, as read from the mount control system.

A fifth (auxiliary) data item is also written: the current rotator angle in degrees.

This data format is one of several accepted by the TPOINT telescope pointing data analysis program: see the TPOINT manual for further details.

3.3.26 tcsdebug

This command sets the TCS debug level. All debug messages at or below the level specified will be output to the system wide debug file tagged with a time stamp and the name of the TCS component to identify these messages from any debug information from another component.

The higher the debug level specified, the more information will be logged. Data created when using a low level of debugging will all be available to the OCS, whereas the data created with a higher level of debugging will be only be created locally to the TCS.

The following table presents a summary of the implemented debug levels. The debug information shown is output when the TCS debug level has been set to be greater or equal to the listed level:

TCS Debug Level	TCS Debug Information
0	No logging
1	Name of TCS command issued
2	TCS command parameters
3	Duration of TCS Command Acknowledgment and Execution
4	Name of TCS Command & Acknowledgment Queues. Additional information about Command Message and Queue Handling
5	Guide target calculated X,Y positions (output at 1 Hz)
6	Polynomial coefficients used to calculate position demands (output at 1 Hz)
7	Name of VI that performs a TCS command action Sidereal Time (in radians) when Slow Loop is executed (output once per minute)
9	Mount and rotator position demands as calculated by Fast loop (output at 20 Hz)
10	Ra and Dec demand of current target, from Fast Loop (output at 20 Hz)

3.3.27 setaccessmode

This command specifies whether the TCS is to be used in Operator or Engineering mode. Some commands may only be available when in Engineering mode. A password is required if the access mode is to be set to Engineering and this is supplied as a command parameter.

3.3.28 m1cover

This command is used to open, close or stop the movement of the primary (M1) mirror cover.

3.3.29 rotoffset

The rotoffset command sets the value of the Instrument Alignment Angle (as defined above in section 3.3.3.1).

3.4 Tracking of Planets and Non-Sidereal Objects

A number of methods will be provided in the TCS to allow pointing and tracking of planets and other non-sidereal rate objects.

3.4.1 Specification by object name only (commands: planet, gplanet, wplanet)

The science, guide or WFS target may be specified to be one of the major planets or the Moon by specifying it by name using the command planet, gplanet or wplanet. The position of the planet will be obtained by the TCS using a local copy of the JPL DE405 Ephemeris data file. During tracking, the planet's position will be updated regularly by the TCS.

3.4.2 Specification by position and fixed differential track rate (commands target, gtarget, wtarget)

A fixed differential track rate may be specified for a science, guide or WFS target using the commands target, gtarget or wtarget. See Section 3.3.4.1 for further details of these commands.

3.4.3 Specification by a file of times and positions (command ephemeris)

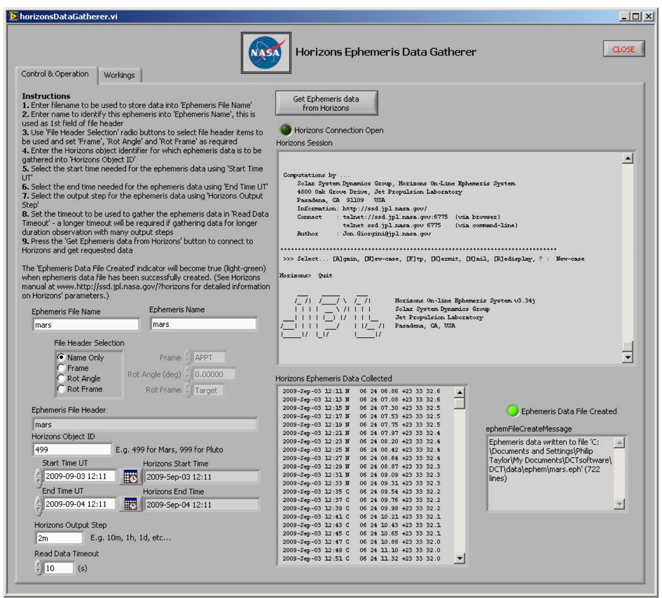
The TCS can read an ephemeris file which defines the TCS Science target's position as a set of RA & Dec values at specified times. Positions are updated regularly by the TCS using this data, interpolating between the supplied data points if required. The differential track rate need not be constant. See Section 3.3.5 for further details of the 'ephemeris' command.

3.4.4 Automated generation of ephemeris files using Horizons

In order to simplify the process of generating suitable ephemeris files, an Ephemeris service utility is provided as part of the TCS. This uses the Telnet protocol to communicate with the JPL Horizons server and is implemented using the LabVIEW Internet Connectivity Toolkit.

The utility can be used at any time to generate an ephemeris file, either before or during observing. It requires a functioning Internet connection to obtain data from the JPL Horizons system.

The user interface is illustrated below. Text in the upper left hand part of the screen provides detailed instructions on how to use the interface.



The data will be written to a file called target.eph. This is the file that can be passed directly to the TCS by specifying the file in the 'ephemeris' command. In the case illustrated, the file produced will be called mars.eph.

Although normally the default settings will be satisfactory, it is possible to change various inputs to the utility as required.

Note that the interface illustrated is subject to revision: in particular the file header options will be removed in a future release as these items will be supplied as parameters with the 'ephemeris' command.

3.5 TCS Status Interface

The TCS will make status data items available as LabVIEW Shared Variables. All status is published in the status variable called TCSTcsStatusSV. Additionally, certain status items are also written to individual shared variables. These are indicated below in the 'Individual SV' column of the table.

A status item is separated out into an individual SV when another subsystem monitors that individual item for change events or alarm status.

3.5.1 Update rates

1. All TCS Public Status data items (in shared variable TCSTcsStatusSV) are updated regularly at a frequency of 1 Hz.
2. Individual status items being monitored by another subsystem for alarm changes are updated regularly at a frequency of 1 Hz. These are indicated by **(A)** in the 'Individual SV' column of the table below.
3. Individual status items being monitored by another subsystem for events are only updated when the value changes. These are indicated by **(E)** in the 'Individual SV' column of the table below.

Status Item	Individual SV (in addition to TCSPublicStatusSV)	Type	Format/Range/Units	Description
State	(E) TCSStateSV	Enum	STANDBY | OFF | ENABLED | DISABLED | FAULT	TCS state
Health	-	Enum	GOOD | WARNING | BAD	TCS health
Heartbeat	-	U32	> 0	1 Hz Incrementing counter
LST	-	String	hh:mm:ss.ss	Current LST as a string
Time	-	LabVIEW Timestamp		Current time (UTC) represented as a LabVIEW timestamp.
targetName	(E) TargetNameSV	String		Name of current Science target
currentHA	-	Double	degrees	Current target Hour Angle (hours)
demandRA	-	Double	degrees	Demanded RA of current target
demandDec	-	Double	degrees	Demanded declination of

				current target
currentRA	-	Double	degrees	Current RA of mount
currentDec	-	Double	degrees	Current declination of mount
demandAz	-	Double	degrees	Demand mount azimuth
demandEl	-	Double	degrees	Demand mount elevation
currentAz	(A) CurrentAzSV	Double	degrees	Current mount azimuth
currentEl	(A) CurrentElSV	Double	degrees	Current mount elevation
currentParAngle	(A) currentParAngleSV	Double	Degrees	Current Parallactic Angle
demandRotPA	-	Double	degrees	Demanded physical rotator angle
currentRotPA	-	Double	degrees	Current physical rotator angle
currentRotIPA	-	Double	degrees	Current rotator Position Angle on the sky
currentRotIAA	-	Double	degrees	Current rotator Instrument Alignment Angle
demandRAstring	-	String	hh:mm:ss.ss	Demanded RA of current target as a sexagesimal string
demandDecstring	-	String	dd:mm:ss.s	Demanded declination of current target as a sexagesimal string
RAstring	-	String	hh:mm:ss.ss	Current RA of mount as a sexagesimal string
Decstring	-	String	dd:mm:ss.s	Current declination of mount as a

				sexagesimal string
demandAzstring	-	String	ddd:mm:ss.s	Demanded mount azimuth as a sexagesimal string
demandElstring	-	String	dd:mm:ss.s	Demanded mount elevation as a sexagesimal string
currentAzstring	-	String	ddd:mm:ss.s	The current azimuth as a sexagesimal string
currentElstring	-	String	dd:mm:ss.s	The current elevation as a sexagesimal string
azCurrentWrap	-	Enum	+1 | 0 | -1 (see 3.3.22 above)	Current azimuth wrap state
rotCurrentWrap	-	Enum	+1 | 0 | -1 (see 3.3.22 above)	Current rotator wrap state
azError	-	Double	arcsecs	Current azimuth error
elError	-	Double	arcsecs	Current elevation error
currentZd	(A) CurrentZD_degsSV	Double	degrees	Current zenith Distance
airmass		Double		Current airmass
Limit Warning and Alarm Items *These items are are individual SVs, not included in TCSTcsStatusSV.*	(E) BlindSpotWarningSV	Boolean	TRUE | FALSE	Elevation max. limit reached in less than limitwarntime minutes
	(E) BlindSpotAlarmSV	Boolean	TRUE | FALSE	Telescope in Blind Spot *or* Elevation max. limit reached in less than limitalarmtime minutes
	(E) ObjectSetWarningSV	Boolean	TRUE | FALSE	Elevation min. limit reached in less than limitwarntime minutes
	(E) ObjectSetAlarmSV	Boolean	TRUE | FALSE	Elevation min. limit reached in less than limitalarmtime minutes

Limit Warning and Alarm Items	(E) AzWrapWarningSV	Boolean	TRUE | FALSE	Azimuth wrap limit reached in less than limitwarntime minutes
	(E) AzWrapAlarmSV	Boolean	TRUE | FALSE	Azimuth wrap limit reached in less than limitalarmtime minutes
	(E) RotWrapWarningSV	Boolean	TRUE | FALSE	Rotator wrap limit reached in less than limitwarntime minutes
	(E) RotWrapAlarmSV	Boolean	TRUE | FALSE	Rotator wrap limit reached in less than limitalarmtime minutes
	(E) AirmassWarningSV	Boolean	TRUE | FALSE	Airmass limit reached in less than limitwarntime minutes
	(E) AirmassAlarmSV	Boolean	TRUE | FALSE	Airmass limit reached in less than limitalarmtime minutes
elZenithLimit	-	Double	degrees	Elevation maximum tracking (blind spot) limit
inBlindSpot	(E) InBlindSpotIsTrueSV	Boolean	TRUE | FALSE	Is telescope currently tracking inside the zenith blind spot?
currentTimeTo Unobservable	(A) CurrentTimeToUnobservableSV	Double	minutes	Time to Current Target unobservable (elevation below minimum)
currentTimeTo Observable	(A) CurrentTimeToObservableSV	Double	minutes	Time to Current Target observable (elevation above minimum)
timeToAzlim	(A) TimeToAzLimitSV	Double	minutes	Time to Az Wrap limit for Current Target

timeToRotlim	(A) TimeToRotLimitSV	Double	minutes	Time to Rotator Wrap limit for Current Target
timeToBlindSpot	(A) TimeToBlindSpot_minSV	Double	minutes	Time before target enters zenith blind spot
timeToBlindSpot Exit	(A) TimeToBlindSpotExit_minSV	Double	minutes	Time before target exits the zenith blind spot
sunDistance	-	Double	degrees	Distance between Sun and Science target
sunProximity	(E) SunProximityTooCloseIsTrueSV	Boolean	TRUE | FALSE	Sun too close to science target?
moonDistance	-	Double	Degrees	Distance between Moon and Science target
moonProximity	(E) MoonProximityTooCloseIsTrueSV	Boolean	TRUE | FALSE	Moon too close to science target?
axesTrackMode	(E) axesTrackModeSV	Enum	AzEl | All	Mode defining the active axes, including which are used to determine the overall inPosition and Tracking status (see 3.6.1 below).
inPosition	(E) InPositionIsTrueSV	Boolean	TRUE | FALSE	All relevant telescope axes are in position and tracking (see 3.6.2 below).
inPositionAz	(E) InPositionAzIsTrueSV	Boolean	TRUE | FALSE	The Azimuth axis is in position and tracking (see 3.6.2 below).
inPositionEl	(E) InPositionAzIsTrueSV	Boolean	TRUE | FALSE	The Elevation axis is in position and tracking (see 3.6.2 below).

inPositionRot	(E) InPositionRotIsTrueSV	Boolean	TRUE | FALSE	The rotator axis is in position and tracking (see 3.6.2 below).
m1CoverState	(E) M1CoverStateSV	Enum	Open | Closed | PartiallyOpen	Primary mirror cover position
mountGuideMode	-	Enum	NoTrack | OpenLoop | ClosedLoop	Guiding mode for telescope mount
accessMode	-	Enum	operator | engineer	Current system access mode
ExternalTargetCfgCmdPreviewIsTrue	(E) ExternalTargetCfgCmdPreviewIsTrueSV	Boolean	TRUE | FALSE	TRUE if external command preview is switched on
NewScienceTarget	(E) NewScienceTargetSV			Compound item which is updated whenever any of its constituent data items change. This item is not included in TCSTcsStatusSV but is in an individual SV.
	1. trackID	Double	(see 3.6.3 below)	Unique Track identifier
	2. demandRA	Double	degrees	Demanded RA
	3. demandDec	Double	degrees	Demanded Dec
	4. pointingOriginX	Double	mm.	Pointing Origin X Co-ordinate
	5. pointingOriginY	Double	mm.	Pointing Origin Y Co-ordinate
	6. demandRotAxisRA	Double	degrees	Demanded RA of Rotator Axis
	7. demandRotAxisDec	Double	degrees	Demanded Dec of Rotator Axis
	8. Contents of class ScienceTargetConfiguration			Class defined in reference [RD8].

3.6 Notes on TCS Status Items

3.6.1 axesTrackMode

The TCS can operate in two different modes which determine which telescope axes are considered active and relevant for status checking. This mode can only be set using a local switch in the TCS - no external command is available. The two settings for this mode are:

- AzEl. Only the azimuth and elevation axes inPosition status will be combined when calculating the value of the overall inPosition status item. The internal TCS overall Tracking status will use only the Tracking status of the azimuth and elevation axes. It is assumed that rotator tracking will be switched off when in this mode (although this is not necessary).
- All. All 3 axes (azimuth, elevation and rotator) inPosition status will be combined when calculating the value of the overall inPosition status item. The internal TCS overall Tracking status will use the Tracking status of all 3 axes.

3.6.2 inPosition

The inPosition flag is set to True when all relevant telescopes axes are tracking 'on target'. The details of how this flag are used is as follows.

The TCS will set inPosition to True when all the "relevant" mount axes are in position. The relevant axes are defined to be either azimuth + elevation only, or all 3 (azimuth, elevation and rotator), depending on the setting of the axesTrackMode (see section 3.6.1 above).

Similar, individual inPosition status flags are also provided for the Azimuth, Elevation and Rotator axes.

When executing a new target or offset command the TCS will initially set inPosition to False. Each axis is defined to be in position when the position error on that axis is less than a specified value (X) for N iterations, where X and N are parameters in the configuration file. Note that changing these parameters currently requires a restart of the TCS . Each iteration period is normally 50 milliseconds (the TCS loop rate is 20 Hz).

Note that for small moves and small values of N, inPosition could go False for a very short time.

3.6.3 trackID

The trackID is one of the data items in the compound item NewScienceTarget, which is updated whenever any of its constituent data items change. The trackID is a unique identifier which is generated whenever a new tracking target is defined (including target position offsets).

The trackID value used is a Double value representing the time the new track was created, expressed as a Modified Julian Date (MJD).

3.7 Mount Control Interface

The main function of the mount control interface is to drive the azimuth, elevation and rotator axes to their desired demand positions and maintain status displays that reflect the current mount state. It provides an interface between the TCS and the mount hardware. Note that we will refer to this part of the TCS system as the Mount Control *Component* (MCC). The term Mount Control *Unit* (MCU) is reserved for the low-level (GDST supplied) system that directly controls the axes motion control hardware.

The TCS pointing kernel will deliver a continuous (20 Hz) stream of time stamped azimuth, elevation and rotator angle demands to the mount control component. If required, this component will interpolate these demands to the resolution required by the mount control unit.

The interface supplied by the GDST-MCU (see [RD7]) to control the mount has two categories of commands:

1. Commands controlling the azimuth and elevation axes.
2. Commands controlling the cassegrain rotator axis.

The interface supplied by the mount control component includes the same command categories plus commands that control all axes, e.g. there is a single stop command that causes all axis movement to stop, in addition to azimuth/elevation stop command and rotator stop command. Note there cannot be an individual azimuth stop command and an individual elevation stop command as such commands are not present in the GDST-MCU interface.

To easily identify the scope of individual MCC commands the following command naming convention is used:

- `mcc<action>` : the command action will govern operation of all axes, e.g. the command `mcctrack` will cause all axes to begin tracking.
- `mccazel<action>` : the command action will govern operation of only the azimuth and elevation axes, e.g. the command `mccazeltrack` will only cause the azimuth and elevation axes to begin tracking.
- `mccrot<action>` : the command action will govern operation of only the rotator axis, e.g. the command `mccrottrack` will only cause the rotator axis to begin tracking.

The commands available under each of the above three categories are detailed in the following sections in command tables. Each command will result in specific commands from the GDST MCU Interface ([RD7]) being sent over the network to control the mount, the MCU commands used by each MCC command are given in the command table column 'MCU Command'. Some of the commands block the receiving of other commands, e.g. the command mccstop blocks and will prevent the acceptance of other commands until it completes, i.e. the mount is stopped or the command times out. Blocking commands are identified in the command tables by a '✓' (tick) in the 'Blocking' column.

3.7.1 MCC control Commands

The table below lists the set of commands that will control operation of azimuth, elevation and rotator axes. These commands are prefixed with the identifier `mcc`.

Command	Parameter	Type	Limits (default)	Blocking	MCU Command	Comments

Command	Parameter	Type	Limits (default)	Blocking	MCU Command	Comments
mccenable				✓	CTAKE	Enable TCS conrol of mount by taking computer control
mccdisable				✓	CREL	Disable TCS control of mount by releasing computer control
mccreference				✓	RFT	Reference the encoder tapes of all axes
mcctrack	state	Enum	ON | OFF	✓	PD, N1D	Turn on/off TCS demand tracking to all axes
mccstop				✓	STOP, NSTOP	Stop all axes movement
mccdebug	level	Enum	0 – 10		–	Set the MCC debug level

3.7.1.1

3.7.1.2 mccenable

Before the TCS can send any commands to the mount, other than to retrieve status, the TCS must take computer control of the mount. This is done using the mccenable command that sends to the MCU the command ‘CTAKE’.

3.7.1.3 mccdisable

Once the TCS has completed its operations (e.g. at the end of night’s observations) it should release computer control of the mount to ensure it does not prevent control being taken by other control systems, i.e. the MCU or Portable Maintenance Unit (PMU). This is done using the mccdisable command that sends to the MCU the command ‘CREL’.

3.7.1.4 mccreference

The mount is accurately positioned using encoders; therefore before the absolute position of each axis can be determined they must be referenced. The MCU interface ([RD7]) contains the command ‘RFT’ to complete this operation, the mccreference command sends to the MCU this command and monitors for the successful tape referencing of each axis to complete. If current status shows all axis encoder tapes are referenced this command will return immediately without sending to the MCU the reference command.

3.7.1.5 mcctrack

The mount tracking state can be on or off. When tracking is on the MCC will direct the stream of position demands from the TCS Pointing Kernel (PK) to the mount. Tracking can only be switched on if the current target was configured and accepted by the TCS within the target-time-to-live period. The target-time-to-live is a TCS configuration parameter and is used to ensure tracking is not switched on using a target configured too long in the past. Each time a target is successfully configured, and sent to the TCS, the time is stored internally in the TCS, when tracking is turned on the stored time is checked with current time against the target-time-to-live, if the time-to-live has expired then turning tracking on will fail with error explaining the cause.

When tracking is switched off the MCC will direct a stream of fixed position demands to the MCU. This fixed position may be as the result of a mccazelmove and/or mccrotmove command, or if tracking has just been turned off, it will be the demand at the point at which the ‘mcctrack off’ command was received. In all of these cases the mount hardware will continue to servo to the demands.

3.7.1.6 mccstop

This command stops all axes using the MCU commands ‘STOP’ (to stop azimuth and elevation) and ‘NSTOP’ (to stop rotator); these commands bring the axes to a controlled stop, power off the drives and apply the brakes.

NB. The MCU command ‘STOP’ does cause all axes to stop, including the rotator. The ‘NSTOP’ command is also sent by the mccstop command to ensure the MCU is explicitly told the rotator should also be stopped.

3.7.2 MCC Azimuth/Elevation control Commands

The table below lists the set of commands that control operation of azimuth and elevation axes. These commands are prefixed with the identifier `mccazel`.

Command	Parameter	Type	Limits (default)	Blocking	MCU Command	Comments
mccazelmove	azimuth (degs)	Double	0.0 – 359.99999		PD	Move the mount to the specified mount az/el position and stop.
	elevation (degs)	Double	-5.0 – 95.0			
mccazeltrack	state	Enum	ON | OFF	✓	PD	Turn on/off TCS demand tracking for azimuth and elevation axes

3.7.2.1 mccazelmove

This command causes the mount to move to a specific azimuth and elevation position and then stop. Note that the positions specified with this command will be passed without modification directly to the mount hardware. No telescope pointing model corrections are applied. The azimuth and elevation values will therefore differ slightly from the topocentric apparent azimuth/elevation frame used with when specifying AZEL for the TCS Target command Frame.

The issue of a mccazelmove command whilst the mount is tracking (see below) will execute an implicit ‘mccazeltrack off’ command

3.7.2.2 mccazeltrack

The azimuth and elevation tracking state can be on or off. When tracking is on the MCC will direct the stream of position demands from the TCS PK to the azimuth and elevation axes using the MCU command ‘PD’. As with mcctrack command, the target-time-to-live is checked when tracking is switched on to ensure it has not expired, see section 3.7.1.5 for details.

When tracking is switched off the MCC will direct a stream of fixed position demands to the MCU. This fixed position may be as the result of a mccazelmove command or if tracking has just been turned off it will be the demand at the point at which the ‘mccazeltrack off’ command was received. In all of these cases the mount hardware will continue to servo to the demands.

3.7.3 MCC Rotator control Commands

The table below lists the set of commands that control operation of rotator axes. These commands are prefixed with the identifier `mccrot`.

The set of rotator commands corresponds closely to the altitude and elevation commands described above.

Command	Parameter	Type	Limits (default)	Blocking	MCU Command	Comments
mccrotmove	angle (degs)	Double	0.0 – 359.99999		N1D	Move the rotator to the specified angle and stop.
mccrotstop				✓	NSTOP	Stop all rotator movement
mccrottrack	state	Enum	ON | OFF	✓	N1D	Turn on/off TCS demand tracking for rotator axis

3.7.3.1 mccrotmove

This command causes the mount to move the rotator to a specific position and then stop. Note that the angle specified with this command will be passed without modification, directly to the mount hardware. No telescope pointing model corrections are applied.

The issue of a mccrotmove command whilst the mount is tracking (see below) will execute an implicit 'mccrottrack off' command

3.7.3.2 mccrotstop

This command stops all rotator axis movement using the MCU command 'NSTOP', this command brings the axis to a controlled stop, powers off the drives and applies the brakes.

3.7.3.3 mccrottrack

The rotator tracking state can be on or off. When tracking is on the MCC will direct the stream of position demands from the TCS PK to the rotator axis using the MCU command 'N1D'. As with mcctrack command, the target-time-to-live is checked when tracking is switched on to ensure it has not expired, see section 3.7.1.5 for details.

When tracking is switched off the MCC will direct a stream of fixed position demands to the MCU. This fixed position may be as the result of a mccrotmove command or if tracking has just been turned off it will be the demand at the point at which the 'mccrottrack off' command was received. In all of these cases the mount hardware will continue to servo to the demands.

3.7.4 MCC Tracking Demands

As mentioned above, when tracking the TCS will produce a 20 Hz stream of time-stamped position demands for each axis, along with the desired velocities. The time stamp will specify the time at which the mount axes should be at the demanded position; this will be passed to the mount in the 'time' parameter of the 'Position Designate' MCU commands ([RD7]). The TCS will produce a new demand every 50 ms, and these demands will be fully corrected to allow for the calibrated pointing model corrections, differential track rates etc.

The following definitions are based on the information contained in the DCT MCU Software Interface document [RD7].

3.7.4.1 Azimuth/Elevation Tracking

The telescope mount will track using 'Position Designate Mode' for the Azimuth and Elevation axes, using MCU command 'PD'. The TCS knows the required positions for a specified time in the near future and so will provide the position, velocity and time stamp with the command. The data stream for the Azimuth/Elevation axes tracking consists of the following data:

Item	Type	Range	Description
azp	Double	0.0 to 359.99999	Demanded azimuth at specified time
elp	Double	-5.0 to +95.0	Demanded elevation at specified time
time	Double	0.0 to 86400.0	UTC time of day (in seconds) corresponding to the specified positions
cw	Enum	C I W I S	Desired cable wrap: Clockwise, Counter-clockwise or Shortest
azvel	Double	-20.0 to +20.0 deg/s	Desired azimuth velocity
elvel	Double	-20.0 to +20.0 deg/s	Desired azimuth velocity

3.7.4.2 Rotator Tracking

The telescope's Cassegrain rotator will track using 'CASS Position Designate Mode', using MCU command 'N1D'. The TCS knows the required position for a specified time in the near future and so will provide the position, velocity and time style with the command. The data stream for the Cass rotator tracking consists of the following data:

Item	Type	Range	Description
Np	Double	0.0 to 360.0	Demanded rotator angle at specified time
cw	Enum	C I W I S	Desired cable wrap: Clockwise, Counter-clockwise or Shortest
time	Double	0.0 to 86400.0	UTC time of day (in seconds) corresponding to the specified positions
Nvel	Double	-20.0 to +20.0 deg/s	Desired rotator velocity

3.7.4.3 Timestamping

In addition to the time sent, corresponding to the demanded position and velocity, the TCS will also keep an internal timestamp with each demand, showing the time that the demand was sent.

3.7.5 Tracking Demand Timing

The TCS will be using the MCU in Position Designate Mode, in which the position and velocity are sent along with the time corresponding to the demand (as shown in the previous sections).

The stream of demands will be sent at a 20Hz frequency i.e. 50 ms between tracking demands. Each position/velocity demand will be sent with a timestamp corresponding to the current time, with velocity calculated based upon a 50 ms look-ahead. The MCU uses an internal 3 slot buffer in this mode and interpolates or extrapolates velocity demands as required. Further discussion of this topic is contained in the Software Design Document ([RD3]).

To ensure a smooth trajectory the spacing between the times of arrival for the tracking demands should not vary excessively. The target jitter for the timing of the tracking command stream is ±10 ms, such that the interval between successive demands should not be outside the range 40-60 ms.

3.7.6 Mount Status

Three different types of status are requested by the MCU, at different frequencies, as follows:

1. The TCS will be using the MCU in Position Designate Mode, sending a MCU 'PD' and 'N1D' command every 50 ms. The MCU response to the PD command contains the current azimuth and elevation and the corresponding position errors. The MCU response to the N1D command contains the current rotator position and position error. The current position of all 3 axes will therefore be available to the TCS at a frequency of 20 Hz.
2. A more general MCU status request will be sent by the TCS once every second. The MCU 'Time Tagged Status Request' (command 'SS') will be used to retrieve this status. This command returns axes and cable wrap positions with timestamp, current operational mode and information on faults.
3. A full status (command 'FB') will be issued every 5 seconds, The FB command is issued with 3 different parameters values and these are cycled through such that the same status request will be issued every 15 seconds. The first type of FB command retrieves the MCU status, the second the AzEl PDU (Power Distribution Unit) status, and the third the Cass Rotator PDU status.

For details of these status commands, refer to the relevant sections of GDST MCU Software Interface ([RD7]).

3.8 Guider Control System (GCS) Interface

The TCS runs a Mount Virtual Telescope that generates the required demand positions for the Alt and Az mount axes and the rotator. An additional Virtual Telescope is used with the Guide target which takes the RA/Dec, co-ordinate frame, wavelength etc. of the guide target and calculates the expected position of the guide star image. These values are then used to specify the expected guide target position on the focal plane. The TCS command to define the Guide target (gtarget) is shown above in section 3.1.

The autoguider measures the actual (X,Y) of the guide star image and, knowing where the image is supposed to be, calculates the difference between the predicted and measured positions. These values are sent using a shared variable to the TCS, which uses these guider correction values to adjust the telescope pointing model parameters.

When guiding is on, neither the telescope RA/Dec, nor its pointing origin (the position of the science target on the focal plane) is altered in the TCS. The telescope pointing model implements the guiding adjustments using terms called GA and GB, which are telescope collimation adjustments. The effect is to adjust the (X,Y) position of the science target RA/Dec in the focal plane

3.8.1 Guider Positioning and Geometry

The predicted guide target position on the focal plane is sent from the TCS to the GCS and guider corrections are sent from the GCS to the TCS. The co-ordinate frame used for both these values is the same: the "ideal TCS plane" co-ordinate frame. This is attached to the rotating focal plane, with the Y axis aligned with sky position angle zero. Any rotation relative to this frame, or non-linear effects, must be corrected internally within the guider.

(i) The predicted guide target position sent from the TCS to the GCS is expressed in mm on the focal plane, with the zero point being the current pointing origin.

(ii) The guider corrections sent from the GCS to the TCS are expressed as a ± offset in mm, with the zero point corresponding to the predicted target position.

As mentioned above, the TCS pointing kernel uses guiding terms called GA and GB, which are telescope model adjustments. With an altazimuth mount, collimation corresponds to azimuth and pitch to elevation. The TCS Guide component will transform the received guider X offset value to collimation (GA) and the guider Y offset value to a pitch value (GB) using the current rotator angle.

3.8.2 Guider-TCS Interaction

The wavelength of the guider target may change, for example when the guider filter is changed. This will affect the predicted positions of the guide probes and so the command 'gwavelength' should be sent to the TCS from the GCS whenever the guide target effective wavelength changes.

The GCS will be informed whenever a new Science target is defined in the TCS (see section 3.8.5.2 below) and the current science target trackID will be sent along with each predicted guide target position.

Guiding corrections will only be implemented by the TCS when guiding has been switched on in the TCS ('guide' command issued with parameter 'on') and the telescope azimuth and elevation axes are tracking.

All guiding correction values received by the TCS will be used, as long as they are in range and the trackID received with the guiding correction corresponds to that of the current Science target. A guide correction timestamp (corresponding to the mid-point of the guiding correction measurement) will be sent from the GCS to TCS and a warning will be raised by the TCS if the data is too old.

3.8.3 Multiple Guiders

The GCS will provide guiding correction values which might be provided by more than one guide probe. This specific guide probe being used is not of interest to the TCS: it uses the provided guiding corrections values, regardless of their source.

In some cases, explicit support of multiple guiders will require appropriate data to define which guiding source is being used to supply the TCS guiding corrections. The guiderName item has been provided for this purpose.

3.8.4 TCS/GCS Command and Data Interface

The following table defines the command and data items used to transfer data between the TCS and Guider systems. The table shows only TCS/Guider interface items that are not defined elsewhere: see section 3.8.5 below for TCS Commands and Status used by the Guider.

3.8.4.1 Data for GCS Published by TCS

Command	***Data Item Name***	***Type***	***Units***	***Valid Set/Range (default)***	***Description/Notes***
guideTargetPosition	gtargX gtargY	Double Double	mm mm	-1000.0 - +1000.0 -1000.0 - +1000.0 (Default: 0.0)	Expected Guide Target position in ideal TCS (X,Y) focal plane co-ordinates. Origin is the current pointing origin. These data items will be updated by the TCS at 20 Hz.
	trackID	Double			Current science target TrackID. This data item will also be updated by the TCS at 20 Hz.
	gtargTime	LabVIEW Timestamp			UTC time of the time for which the positions were computed.

3.8.4.2 Data from GCS Subscribed by the TCS

Command	***Data Item Name***	***Type***	***Units***	***Valid Set/Range (default)***	***Description/Notes***
guiderMeasuredError	guideX guideY	Double Double	mm mm	-300.0 - +300.0 -300.0 - +300.0 (Default: 0.0)	Guiding correction offsets in ideal TCS (X,Y) focal plane co-ordinates, relative to the predicted guide target position. The values are updated regularly by the Guider at a rate of approximately 1 Hz.
	trackID	Double			TrackId of the target for which guiding corrections are being measured

	guideCorrectionTime	LabVIEW Timestamp			UTC time of the mid-point of the guiding correction measurement
guiderName	name	String			Identifier for the guider currently in use

3.8.5 Commands and Status from the TCS Component used by the GCS

The GCS will make use of TCS component commands and status items that are defined elsewhere. These are as follows:

3.8.5.1 Data for TCS Published by GCS

The **gTargetConfig** command is used to define:

- The RA and Dec of the Guide target (see table in section 3.2.2).
- The Guider target effective wavelength (see table in section 3.2). The command parameter is the wavelength in microns (type Double). This command should be sent to the TCS whenever a new guider target is defined or a guider filter changes.

3.8.5.2 Data for GCS Published by TCS

These status items are defined in the TCS status table above (Section 3.5).

- **inPosition**. The guider should start locking on to a newly acquired science target when this flag takes the value True.
- **NewScienceTarget.** This contains data defining the position of a new science target, the current TCS pointing origin and the new trackID.

3.8.5.3 GCS related TCS Commands

The following TCS commands, defined above in the table in Section 3.2.3, control the use and clearing of guiding correction data by the TCS:

- **guide**. Switches TCS guiding on or off. See section 3.3.18 for further details.
- **guideclear**. Set current TCS guiding corrections to zero. See section 3.3.19 for further details,
- **guideautoclear.** Switch automatic clearing of TCS guiding corrections on or off. See section 3.3.20 for further details.

Wavefront Sensor (WFS) Interface

The TCS runs a Mount Virtual Telescope that generates the required demand positions for the Alt and Az mount axes and the rotator. An additional Virtual Telescope is used with the Wavefront Sensor target which takes the RA/Dec, co-ordinate frame, wavelength etc. of the WFS target and calculates the expected WFS target position in the focal plane. The TCS command to define the WFS target (wtarget) is shown above in section 3.1.

3.8.6 WFS Geometry

The predicted WFS target position on the focal plane is sent from the TCS to the WFS. The co-ordinate frame used this value is the "ideal TCS plane" co-ordinate frame. This is attached to the rotating focal plane, with the Y axis aligned with sky position angle zero. Any rotation relative to this frame, or non-linear effects, must be corrected internally within the WFS subsystem.

3.8.7 TCS/WFS Command and Data Interface

The following table defines the command and data items used to transfer data between the TCS and WFS systems. The table shows only TCS/WFS interface items that are not defined elsewhere: see section 3.8.8 below for TCS Commands and Status used by the WFS.

3.8.7.1 Data for WFS Published by TCS

Command	*Data Item Name*	*Type*	*Units*	*Valid Set/Range (default)*	*Description/Notes*
wfsTargetPosition	wtargX wtargY	Double Double	mm mm	-1000.0 - +1000.0 -1000.0 - +1000.0 (Default: 0.0)	Expected WFS Target position in ideal TCS (X,Y) focal plane co-ordinates. Origin is the current pointing origin. These data items will be updated by the TCS at 20 Hz.
	trackID	Double			Current science target TrackID. This data item will also be updated by the TCS at 20 Hz.
	wtargTime	LabVIEW Timestamp			UTC time of the time for which the positions were computed.

3.8.8 TCS Commands and Status used by the WFS

The WFS will make use of TCS commands and status items that are defined elsewhere. These are as follows:

3.8.8.1 TCS Commands sent from the WFS to TCS

The **wTargetConfig** command is used to define:

- The RA and Dec of the WFS target (see table in section 3.2.2).
- The WFS target effective wavelength (see table in section 3.2). The command parameter is the wavelength in microns (type Double). This command should be sent to the TCS whenever a new WFS target is defined or a WFS filter changes.

3.9 Active Optics System (AOS)

The Active Optics subsystem maintains the optical quality of the telescope by controlling the figure and position of the primary and secondary mirrors. It can operate in both open-loop mode and closed loop mode, using WFS data. For more details, see the section describing the AOS subsystem in the DCT Software Design Document.

The coma correction performed by M2 tilt will contain both the tilt and decenter components. A pointing correction is sent to the TCS so that the spot is maintained at the current offset position. The AOS coma pointing errors will be calculated at the AOS closed loop bandwidth and the TCS reads a new value whenever it is updated.

The X and Y values are sent from the AOS to the TCS in units of arcseconds. The optical coordinate system used by the AOS is fixed with the M1 mirror and rotates with it. The origin of the Optical coordinate system is at the M1 vertex. The Z-axis coincides with the M1 optical axis and is positive in the direction of telescope pointing. The X-axis coincides with the elevation axis. The positive Y-axis is towards Zenith when the telescope is Horizon pointing.

The unmodified X and Y values received from the AOS can therefore be used to adjust the TCS pointing model: they directly correspond to the pointing model terms GA and GB (telescope collimation adjustments). The effect is to adjust the (X,Y) position of the science target RA/Dec in the focal plane.

The TCS will accept pointing offsets from the AOS regardless of whether guiding is switched on. The values received from the AOS are applied in the TCS by adding them to the current total guider correction.

3.9.1 Data from AOS Subscribed by the TCS

Command	***Data Item Name***	***Type***	***Units***	***Valid Set/Range (default)***	***Description/Notes***
aosPointingOffset	aosX aosY	Double Double	arcsecs arcsecs	-300.0 - +300.0 -300.0 - +300.0 (Default: 0.0)	AOS correction offsets in ideal TCS (X,Y) co-ordinates The values are updated regularly by the AOS. Rate TBC.

3.10 Atmospheric Dispersion Corrector (ADC)

The Atmospheric Dispersion Compensator (ADC) corrects for atmospheric dispersion due to the variation in atmospheric refraction with wavelength. The dispersion is worst with wide-band observations at large zenith distances.

No information is currently available regarding the optical design of the DCT ADC, but it may use a pair of counter-rotating prisms which are adjusted to suitable angles to provide the required dispersion correction. The ADC control system would control the prisms.

The setting of the ADC depends on the amount of dispersion to be corrected. When the ADC is in use, the TCS shall regularly provide the current dispersion correction information to the ADC system, to enable calculation of the ADC settings.

3.10.1 Dispersion information sent from TCS to ADC

Atmospheric dispersion is defined as the difference between the atmospheric refraction at a pair of wavelengths - the given and reference wavelengths. The values for these wavelengths depend on how wide a wavelength range is to be corrected for and the characteristics of optics and filter being used. One of the values might be the current science target wavelength. The TCS will send the value of the current atmospheric dispersion that is to be corrected by the ADC.

Depending on the required performance and with some assumptions and simplifications, it might be possible to calculate an accurate enough averaged atmospheric dispersion value that depends only the telescope zenith distance.

A more rigorous approach is for the TCS to calculate the atmospheric dispersion between a specified pair of wavelengths at the current zenith distance and supply this information to the ADC. The direction of dispersion is always towards the zenith and this may also need to be taken into account, depending on the ADC configuration.

3.10.2 ADC and Telescope Pointing

The effect (if any) of the ADC on telescope pointing will depend on the optical design of the ADC. It may involve a change in plate scale and/or offset in image positions. The effective wavelength of the science target would also be changed. It may also be appropriate to have separate telescope pointing models for when the telescope is used with and without the ADC.

If the ADC pointing effects are well-enough understood, pointing correction values could be sent from the ADC to TCS in a similar manner to the Guider. However, if the corrections are small and varying slowly, then these could be handled directly by the Guider.

3.10.3 TCS/ADC Command and Data Interface

3.10.3.1 Data for ADC Published by TCS

Parameter or Data Item Name	*Type*	*Units*	*Valid Set/Range (default)*	*Description/Notes*
wavelength1	Double	microns	0.3 – 10.0	First ('reference') wavelength for dispersion calculation. Default: Current science target wavelength
wavelength2	Double	microns	0.3 – 10.0 (0.6)	Second wavelength for dispersion calculation.
disp	Double	arcsecs	-100.0 - +100.0 (Default: 0.0)	Current atmospheric dispersion

3.10.3.2 Data from ADC Subscribed by the TCS

Command	*Data Item Name*	*Type*	*Units*	*Valid Set/Range (default)*	*Description/Notes*
adcPointingOffset	adcX adcY	Double Double	arcsecs arcsecs	-1000.0 - +1000.0 -1000.0 - +1000.0 (Default: 0.0)	ADC correction offsets in ideal TCS (X,Y) co-ordinates The values are updated regularly by the ADC.
wavelength	wavelength1	Double	microns	0.3 – 10.0 (Default: Current science target wavelength)	First ('reference') wavelength for dispersion calculation.
	wavelength2	Double	microns	0.3 – 10.0 (Default: 0.6)	Second wavelength for dispersion calculation.

3.11 IERS component

The IERS component supplies and updates the IERS parameters that correct for polar motion, UT1 and occasionally leap seconds, which all affect the pointing of the telescope. The IERS component automatically handles obtaining these parameters but leaves the decision as to when they should be applied to the operator.

3.11.1 IERS parameters

The IERS parameters of interest to the TCS are the corrections for polar motion which reflect the offset between the Celestial Ephemeris Pole and the IERS Reference Pole along with the correction to UTC that provides a value for Universal Time (UT1) needed in the calculation of sidereal time. Although predictions can be made for these corrections, ultimately they must be measured. The values are published daily by the IERS and can be downloaded from their website at http://maia.usno.navy.mil/ser7/finals.daily.

3.11.2 Leap seconds

Leap seconds are introduced occasionally to keep civil time (UTC) in step with mean solar time. The last occasion at which a leap second was added was December 31 2008. The TCS requires the number of leap seconds whenever it computes UTC as well as consistent values of leap seconds and the value (UT1 – UTC) whenever it transforms from TAI to UT1.

3.11.3 IERS component commands

Command	Parameter	Range/Units	Comment
iersupdate			Update IERS parameters

3.11.3.1 iersupdate

This command instructs the IERS component to make a connection to the IERS service and fetch the parameters delut, xpm and ypm. If the connection is successful, the component updates the initialization file with the new values and re-writes the "updated" parameter with the current time.

In order to handle the rare occurrences of a leap second, the IERS component will check that the newly read value of delut less the current value is less than 0.9s. If it is greater than this then a leap second has occurred and it will additionally update the initialization file as follows

- mjdnls - the MJD of the next leap second will be written as 99999.9
- delat – the number of leap seconds will be incremented by one

Updating the delut, delat and mjdnls parameters must be done together to ensure time scale consistency. Once the values are updated, a event will be generated with the name $(telescope).time.iersupdate. This event will contain the same information as the standard 1 Hz event (see next section) but can be used by any component that needs to know that any of the time parameters have changed value. In particular it will be used by the pointing kernel to update its internal conversion factors. By using a separate event that is only generated by an explicit iersupdate command we ensure that the pointing of the telescope is only modified when the operator expects it.

3.11.4 IERS component internal status

Internal status items maintained by the IERS component are

Status item	Units/Range	Comment

timehealth	value message	The value is GOOD | WARNING | BAD and the message describes the reason
timeactivity	value message	The value is BUSY | ERROR | IDLE and the message describes the reason.
timenls	LabVIEW Timestamp	The time at which the next leap second will occur
delut	Secs	UT1 minus UTC
delat	Secs	The number of leap seconds
xpm	Arcsecs	x component of polar motion
ypm	Arcsecs	y component of polar motion

All these values will be made available internally through an event $(telescope).time which will be generated once per second.

3.11.5 IERS component health and alarms

The IERS component will set its health to bad or warning and set an appropriate health message under the following circumstances

Event	Health	Response
Unable to access initialization file tcs.ini	Bad	Use default leap seconds value of 33. Set xpm, ypm and delut to 0.0
Unable to locate IERS section in tcs.ini	Bad	Use default leap seconds value of 33. Set xpm, ypm and delut to 0.0
Current date > updated + warndays	Warning	Set health to warning with message "IERS parameters not updated for more than <warndays>"
Current date > updated + maxdays	Bad	Set health to bad with message "IERS parameters not updated for more than <maxdays>"
Time card error	Bad	The types of error the card can encounter are TBC and then suitable error messages will be set
Time card warning	Warning	As for errors, the types of warnings the card can encounter are TBC
Unable to connect to IERS service		Log failed connection attempt but leave health unchanged

3.12 Weather Server Component

The weather server component interfaces to the DCT weather station. It uses this data to provide refraction calculations to the pointing model. As these values are essential for the operation of the TCS, a manual mode can be used to provide values when no weather station is available.

Warnings about meteorological values being out of range will generate Health warnings in the TCS.

3.12.1 Weather Server Commands and Data

Command Name (blank if Data only)	*Parameter or Data Item Name*	*Type*	*Units*	*Valid Set/Range (default)*	*Description/Notes*
wsmode	mode	Enum		automatic | manual	Manual or automatic data for refraction calculation
wsset	temperature pressure humidity	Double Double Double	°C mb % RH	Set in config file Set in config file Set in config file	Parameter values are used only when the mode is 'manual'. Defaults and valid ranges for these data items are defined in configuration file parameters, read at TCS startup.
-	temperature	Double	°C	Set in config file	Data is read periodically from the Weather Station when the mode is 'automatic' Defaults and valid ranges for these data items are defined in configuration file parameters, read at TCS startup.
-	pressure	Double	mb	Set in config file	
-	humidity	Double	% RH	Set in config file	

3.12.2 Notes on Weather Server Commands

3.12.2.1 wsmode

The 'wsmode' command is used to switch the specification of the temperature, pressure and humidity between the default 'automatic' mode and 'manual' mode.

In 'automatic' mode the component will periodically read the environmental parameters from the DCT weather station so that they can be used by the TCS pointing model to compute the refraction corrections.

Should the weather server data be unavailable for any reason, 'manual' mode should be selected and the three parameters are then specified manually using the 'wsset' command.

3.12.2.2 wsset

The 'wsset' command is used to manually set values for the temperature, pressure and humidity that the TCS will use in its refraction calculations. Note that these values will only be used if the mode has been set to 'manual'.