Telescope Control Interfaces

K.Wirenstrand, G.Chiozzi

14/05/97

Contents

- Introduction
- TCS Public Interfaces
- Installing and configuring the VLT TCS Packages
- Using the command interface
- Using the C/C++ interfaces
- Building INS panels to monitor TCS

INTRODUCTION

VLT Instrument Control Software has the need of interacting with the VLT Telescope Control Software (TCS)

The VLT Telescope Control Software provides:

- A set of public interfaces to be used for this purpose.
- A TCS Simulation Package to be used during INS Control Software development to test the interface.

This presentation is a step by step guide to:

- The basic functionalities provided by the TCS Public Interface.
- How to develop applications interfacing to TCS.
- How to use the TCS Simulation Package to test applications.

The basic manual referenced here is:

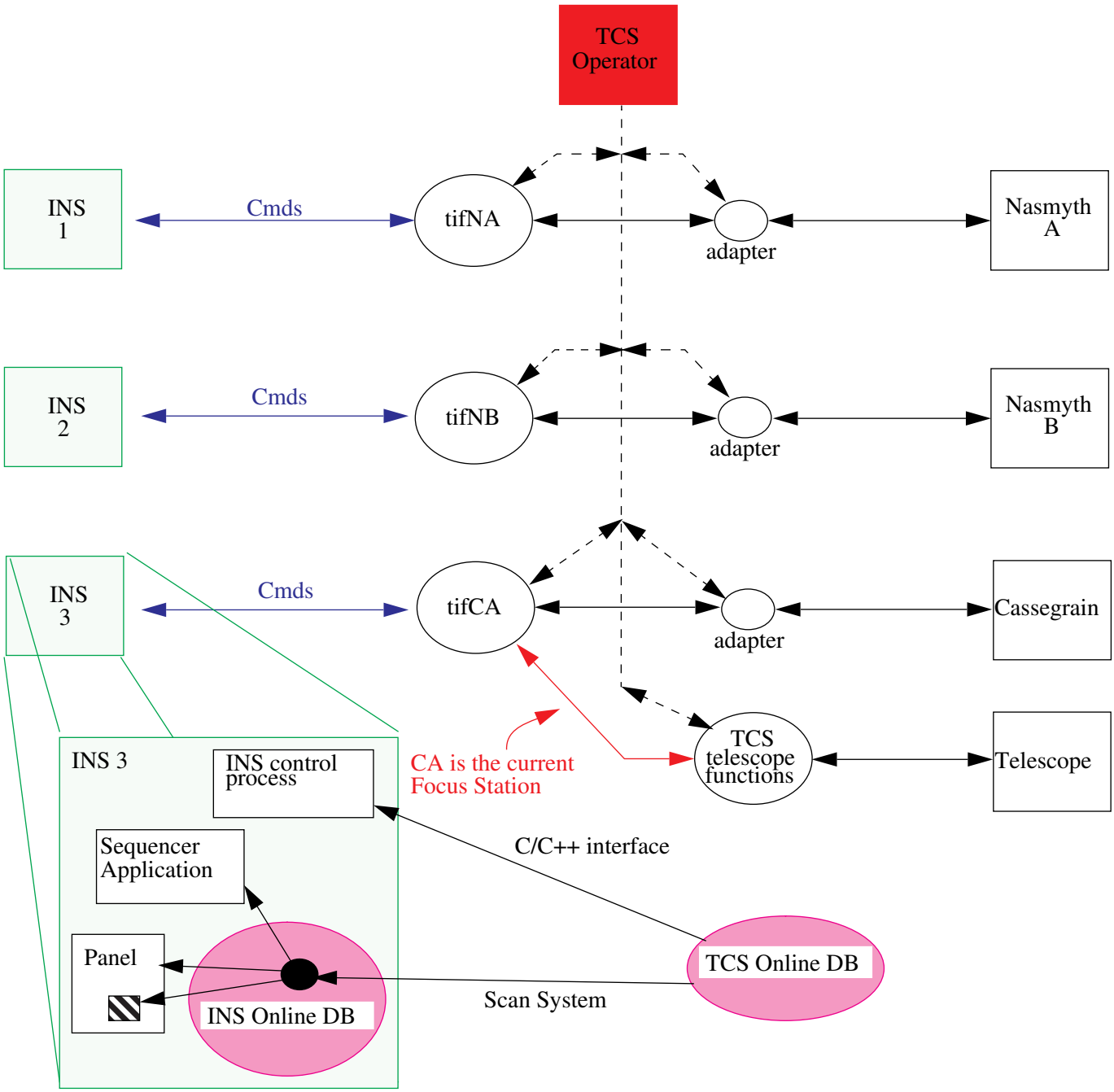
Telescope Control System User Manual, Issue 1.2
VLT-MAN-ESO-17230-0942

TCS Public Interfaces

TCS provides 3 different types of public interfaces (TCS User Manual par. 2.1):

- **Command interface (par. 2.2).**
  It is used mainly to request actions from TCS.
  INS Applications communicates with TCS sending commands to a tif process in the TCS environment.
  - There is one tif process for each Focus Station.
  - The booking system provides protection.
- **C/C++ Interface (par 2.3).**
  It is used by C/C++ Applications to retrieve information from TCS and to get triggered on TCS events.
  It consist of 3 components:
  - Data Query Library (par. 2.3.1)
  - FITS Header Data Functions (par. 2.3.2)
  - Event Configuration Library (par. 2.3.3)
- **Online Database Classes (par 2.4)**
  Used by non C/C++ Applications (Sequencer, Panels) to retrieve information from TCS.
  Panel Editor Classes (par. 2.5) make easier the development of INS panel that also display TCS data.

TCS Public Interfaces (cont.)



Installing and configuring the VLT TCS Packages

The VLT TCS Packages are distributed as part of the VLT Common Software. It currently includes (par. 4.1):

- **the TCS simulation and development package:**
  - all TCS interfaces (libraries, database classes, CDTs) for development.
  - processes and database to simulate a running TCS.
- **the TCS interface to the ESO Star Catalogue Interface**

The TCS Simulation allows to test software that must interface with the VLT TCS, by means of a special database configuration and the tcssim software module.

In order to switch between the communication with a TCS Simulation and the real TCS, it is only necessary to properly set environment variables.

The Installation is clearly described in section 4.6 of the user manual.

... general considerations

The steps to get running a full TCS Simulation are described in section 4.7 of the User Manual.

- The TCS Simulation consists of:
  - A set of TCS Simulation specific processes
  - A set of TCS real processes
  - A TCS Simulation specific database
- The TCS Simulation database can be instantiated in:
  - An independent RTAP environment (suggested)
  - The INS environment itself (in case of a system with limited resources).
- The TCS software can be used both for VLT and NTT: it is very important to specify what telescope we want to use (par. 4.7.2)

..... preliminary steps

Before attempting to use the TCS Simulation it is necessary to:

- Verify general environment variables (par. 4.7.1):
  - RTAPENV
  - INS_ROOT
  - INS_SETUPPATH

- Choose the TARGET_TCS (4.7.2).
- Install the TCS setup and dictionary files in the INS_ROOT running the utility tcssimInstall.sh (par. 4.7.3).

  The procedure creates:
  - the INS_ROOT sub-directories needed by TCS simulation, if not yet existing
  - the TCS Data Dictionary
  - the TCS Alias Tables
  - the default setup file

... getting the TCS Simulation DB running

- **The easiest way to get a complete TCS Simulation environment, is to create and start the test environment for the tcssim module.**
  If the tat system is configured:
  - go in ..../tcssim/test/
  - issue the command tat -nc makeEnv
  - look at the created environment name in the file .testSession
- To add the TCS Simulation branch to an environment (par. 4.7.4):
  - set to the environment dbl directory
  - add the TCS simulation branch to the database:
    edit DATABASE.db file and
    - move to application definitions section.
    - Add the following lines:
      #include “tcssimUser.db”
  - add in the Makefile the definitions for the TARGET_TCS
  - regenerate the (RTAP) database files:
    - $ make clean db
- Once the name for the TCS DB is known, set to it the environment variable **TCS_ENVNAME. This must be set for every process interfacing to TCS (par. 2.6.2).**

.... starting the TCS Simulation

Once the TCS Simulation DB is running and the environment is properly configured, it is possible to start the simulation processes.

Section 2.6 of the User Manual describes in details how to control the simulation (start, stop, operator commands) and the levels of simulation provided.

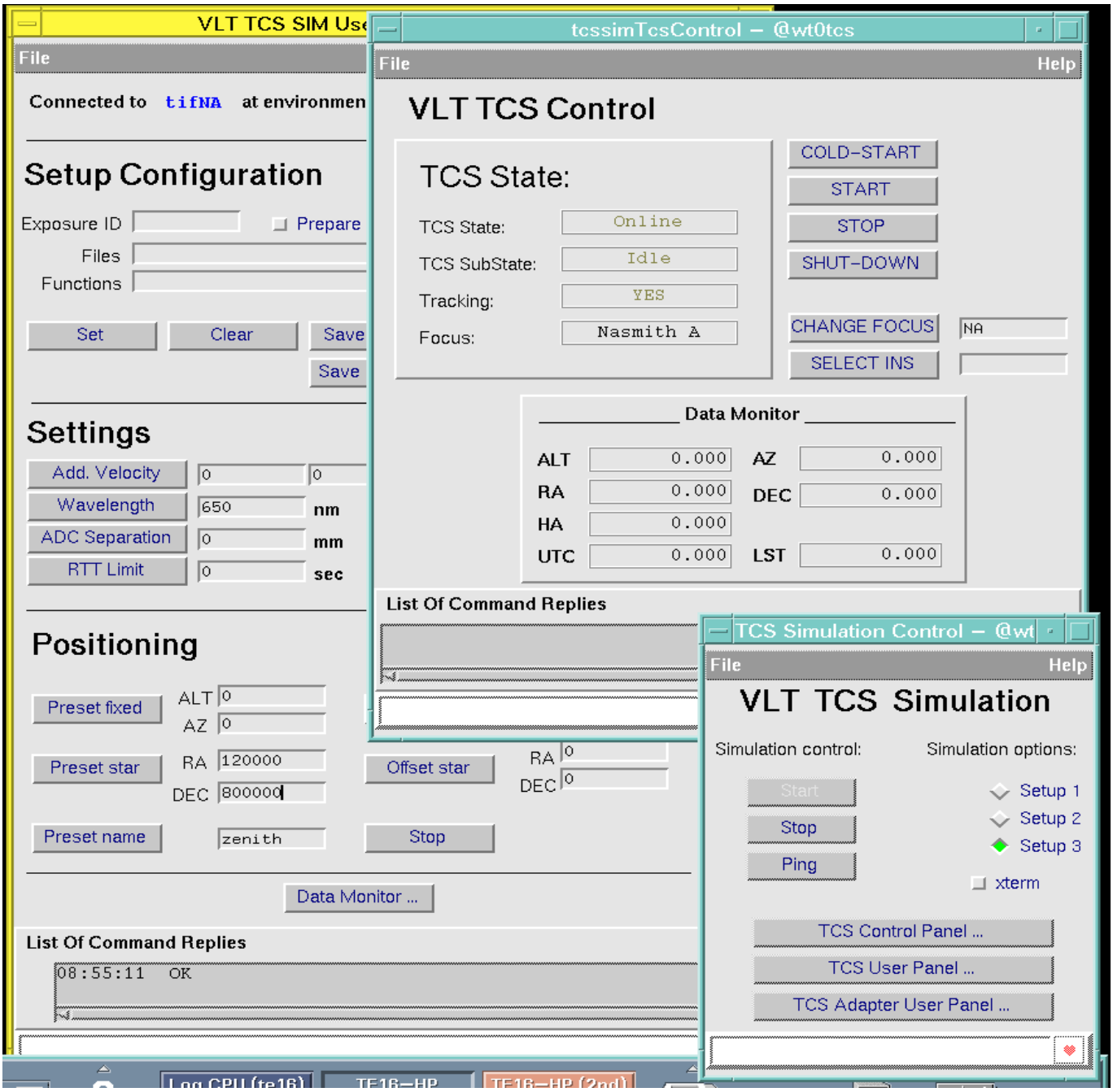
The following setups are delivered:

- Setup 1: simulates the TCS Interface by Command Manager, no TCS module simulation at all
- Setup 2: uses real tif, simulates all other TCS Modules by Command Manager
- Setup 3: uses all existing modules, i.e. simulates as much functionality as possible

The TCS Simulation is easily operated via the control panels:

- The Simulation Control Panel to select, start, stop and check simulation
- The TCS Operator Panel to send basic operator commands.
- An example of a Telescope User Panel, i.e. a panel providing all user commands currently available for telescope control.
- An example of an Adapter User Panel.
- The Data Monitor Panel displays the position and time data which can also be acquired by the Data Query Library.

TCS Simulation Control Panels



Using the command interface

Any application capable of sending commands via the CCS message system can send commands to TCS.

The commands must be sent to the tif process corresponding to the desired focus station (tifNA, tifNB, tifCA).

The CCS Booking System is responsible for access control and security.

Section 2.2 of the User Manual describes the available commands (more details are provided in the specific man pages in section 3.2).

The command definition table (par. 5.4) provides a detailed description of command parameters.

From the instrument point of view the most important command
is probably the command:

SETUP

It is used to PRESET the telescope to a new target, setting at the same time all configuration parameters and support subsystems (Auto Guiding, Active Optics and so on).

No action is actually executed unless a new target is provided in the SETUP command.

The list of setup keywords interpreted by TCS with the command SETUP is described in the default setup file in section 5.5.2.

Available commands

Command	Description
CLRSTP	Clear ready SETUP file with default values
GETINS	Get currently selected instrument
GETINSD	Get instrument configuration parameters
OFFSAA	Issue an ALT/AZ offset step
OFFSAD	Issue an RA/DEC offset step
OFFSADG	Issue an RA/DEC combined offset step
OFFSFOC	Issue a focus offset correction to M2
OFFSXY	Issue an offset step in defined X/Y coordinates.
PRSALAZ	Preset telescope to given ALT/AZ position (fixed)
PRSCOOR	Preset telescope to given RA/DEC coordinates (tracking)
PRSNAME	Preset telescope to named fixed position
SAVCSTP	Save a copy of the current setup file with the given name
SAVRSTP	Save a copy of the ready setup file with the given name
SETADC	Set ADC separation
SETAV	Set additional velocity
SETLAM	Set observed wavelength
SETINSD	Set instrument configuration parameters
SETRLIM	Set limit value for remaining tracking time warning
SETUP	Define a new setup configuration (standard format, as defined in [7], for FITS keywords see section 5.5)
STOPTRK	Stop tracking and stay at actual position

Using the C/C++ interfaces

The TCS public programmatic interface provides a set of C/C++ library. They can be used to retrieve TCS status and data.

The following functions are available (par 2.3):

- TCS query routines (2.3.1):
  - tifGetByName(): general routine to query data items by name
  - tifGetPosition(): the telescope position in various coordinates.
  - tifGetStates(): the states of TCS modules.
  - tifGetStatus(): the status of the TCS

- TCS FITS headers routines (2.3.2):
  - tifGetFitsStart(): Generate the TCS FITS keywords at start of exposure
  - tifGetFitsEnd(): Generate the TCS FITS keyword at end of exposure and dump in file

- TCS Events Handling (2.3.3):
  - tifAttachEvent(): attach to a TCS event
  - tifDetachEvent(): detach from a TCS event

TCS Named Data Items

Name	Type	Unit	Description
LST	vltDOUBLE	HMS	Local Sidereal Time Format: HHMMSS.F, Range: 0 - 240000.0
UTC	vltDOUBLE	MJD	Universal Time Coordinated as modified julian date
HA	vltDOUBLE	HMS	actual Hour Angle Format: HHMMSS.F, Range: 0 - 240000.0
RA	vltDOUBLE	HMS	actual Right Ascension (apparent places 'now') Format: HHMMSS.F, Range 0 - 240000.0
DEC	vltDOUBLE	DMS	actual Declination (apparent places 'now') Format: +/-DDMMSS.F, Range: -900000.0 - 900000.0
RA2000	vltDOUBLE	HMS	actual Right Ascension at mean place of J2000 Format: HHMMSS.F, Range 0 - 240000.0
DEC2000	vltDOUBLE	DMS	actual Declination at mean place of J2000 Format: +/-DDMMSS.F, Range: -900000.0 - 900000.0
ALT	vltDOUBLE	deg	actual Altitude position Range: 0.0 - 90.0
AZ	vltDOUBLE	deg	actual Azimuth position Range: -180.0 - 360.0 (South=0, Est=90)
AIRMASS	vltDOUBLE	none	actual airmass
PRLTIC	vltDOUBLE	deg	actual parallactic angle Range: 0.0 - 360.0
DVELRA	vltDOUBLE	arcsec / sec	actual differential velocity in RA with respect to normal sidereal rate, Range: -15.0 - +15.0
DVELDEC	vltDOUBLE	arcsec / sec	actual differential velocity in Declination with respect to normal sidereal rate, Range: -15.0 - +15.0
LONGI	vltDOUBLE	rad	telescope site Longitude, west is positive
LATI	vltDOUBLE	rad	telescope site Latitude, south is negative
LEVEL	vltDOUBLE	m	telescope site Height above sea level
REMTIME	vltDOUBLE	sec	actual remaining tracking time
REMLIMIT	vltDOUBLE	sec	limit value for event triggering of remaining tracking time

Name	Type	Unit	Description
TRACK	vltLOGICAL	BOOL	flag, indicating if currently in tracking state
FOCUS	vltINT32	focus ID	Current focus station. Can be one of (see mswDefines.h): mswUNDEFINED_FOCUS 0 mswNASMITH_A_FOCUS 2 mswNASMITH_B_FOCUS 3 mswCASSEGRAIN_FOCUS 4 mswCOUDE_FOCUS 5 mswINTERMEDIATE_D_FOCUS 6
TELESCOP	vltBYTES20	string	Telescope name
VERSION	vltBYTES80	string	Version of TCS software
OPER	vltBYTES32	string	Name of telescope operator
FOCU_VALUE	vltDOUBLE	mm	Position of mirror 2 in z-axis
DOME_STATUS	vltINT32	status ID	Status of dome. Always 0 in the current VLT implementation. See NTT specific documentation for NTT values.
AG_STATUS	vltINT32	status ID	Autoguiding substate (see agwsDefines.h): evhSTATE_UNK 0 agwsSTATE_IDLE 7 agwsSTATE_ERROR 6 agwsSTATE_OPERATING 121 agwsSTATE_GUIDING 122 agwsSTATE_MANUAL 123
AG_RA	vltDOUBLE	HMS	actual Right Ascension for current Guide Star (J2000 coordinates) Format: HHMMSS.F, Range 0 - 240000.0
AG_DEC	vltDOUBLE	DMS	actual Declination for current Guide Star (J2000 coordinates) Format: +/-DDMMSS.F, Range: -900000.0 - 900000.0
ROT_ANGLE	vltDOUBLE	deg	Rotator angle in Degrees (DD.ddd)
PROBE_X	vltDOUBLE	mm	Guide probe X coordinate. Always 0 in the current VLT implementation. See NTT specific documentation for NTT values.
PROBE_Y	vltDOUBLE	mm	Guide probe Y coordinate. Always 0 in the current VLT implementation. See NTT specific documentation for NTT values.

FITS Header Keywords

FITS Keyword	Unit	Description
TELESCOP	string	Telescope name
RA	degrees	Right Ascension in mean J2000 (DD.ddd)
DEC	degrees	Declination in mean J2000 (DD.ddd)
EQUINOX	years	Always 2000
RADECSYS	string	Always FK5
LST	seconds	Local Sidereal Time. Seconds since midnight
UTC	seconds	Universal Time. Seconds since midnight
TEL DID	string	Data dictionary name for TCS
TEL DATE	string	TCS release version and date
TEL ALT	degrees	Altitude angle at start
TEL AZ	degrees	Azimuth angle at start (S=0, W=90 !!!)
TEL GEOELEV	meters	Telescope elevation above sea level
TEL GEOLAT	degrees	Telescope geographic latitude (+=North)
TEL GEOLON	degrees	Telescope geographic longitude (+=East !!!)
TEL OPER	string	Name of telescope operator
TEL FOCU ID	string	Focus station (NA, NB, CA or CO)
TEL FOCU VALUE	mm	M2 absolute position in Z-axis
TEL PARANG END	degrees	Parallactic angle at end
TEL PARANG START	degrees	Parallactic angle at start
TEL AIRM END		Airmass at end
TEL AIRM START		Airmass at start
TEL TRAK STATUS	string	OFF, NORMAL or DIFFERENTIAL
TEL TRAK RATEA	arcsec/ sec	Absolute tracking rate in RA
TEL TRAK RATED	arcsec/ sec	Absolute tracking rate in DEC
TEL DOME STATUS	string	FULLY-OPEN, PART-OPEN, CLOSED or VIGNETTING
ADA GUID STATUS	string	ON or OFF
ADA GUID RA	deg	Guide star RA in J2000 (only if GUID ON)
ADA GUID DEC	deg	Guide star DEC in J2000 (only if GUID ON)

Using the libraries

In order to use the libraries, it is necessary:

- To have TCS Simulation and Development Package installed and configured (see 4.6 and 4.7)
- Environment variables properly set (see in particular 4.7.1)
- **A running environment with TCS or TCS Simulation database branch (see in particular 4.7.4)**
- The code should be written in C++ or at least compiled with the C++ compiler, i.e. the files should have the .C (capital C) filename extension.
- The Makefile must contain proper definitions for the TARGET_TCS to be built (see section 4.7.2)
- The following libraries must be linked with the code:
  tifLibs msw tcs tcsVcc oslx slx misc evh eccs fnd C++ CCS
- The include file tif.h contains all the necessary definitions and function prototypes.

Section 5.1 contains three complete and working sample programs, two using the Data Query Library and the other one using the Event Configuration Library.

The second Data Query Library example demonstrates how to use the tifGetFitsStart() and tifGetFitsEnd() to generate a complete FITS header.

Building INS panels to monitor TCS

Two TCS Public Database classes have been implemented to provide access to TCS data for UIF instrument panels and Sequencer scripts.

The two classes have identical structure and differ only in the way they internally access TCS information (par. 2.4):

- tifTCS_PUBLIC can be instantiated in any user (instrument) database to provide a local image of the TCS public data items. This class is used when the TCS database, real or simulated, is NOT in the same environment as the user (instrument) database. In this case, the scan system is used to update the user database with values from the TCS environment (real or simulated).

- tifTCS_LOCAL provides the same items, but can be instantiated in the same environment where TCS itself is running, i.e. when simulated TCS data are a part of the user (instrument) database. In this case, the items are updated using the calculation engine, instead of the scan system.

The tifTCS_PUBLIC DB class

tifTCS_PUBLIC	
tcsState	Global TCS state information
tcsState	TCS global state, as in Mode Switching
tcsSubstate	TCS substate, as in Mode Switching
track	TCS tracking flag (0/1)
focus	Current focus ID
domeStatus	Current dome Status
times	Time items
lst	Local Sideral Time
utc	Universal Time Coordinated
coord	Target coordinates
ha	Target actual Hour Angle
ra	Target actual RA
dec	Target actual declination
ra2000	RA at mean place of J2000
dec2000	dec at mean place of J2000
alt	Telescope altitude
az	Telescope azimuth
prltic	Actual parallactic angle
track	Tracking parameters
airmass	Actual airmass
dvelra	Differential velocity RA
dveldec	Differential velocity dec
rotAngle	Actual rotator angle
remtime	Remaining tracking time
remlimit	Remaining tracking time limit
trackingLost	Event flag for lost tracking
remTimeLow	Event flag for low remaining trk. time
guide	Auto Guide parameters
status	Auto Guide status
ra	RA of current guide star
dec	dec of current Guide Star
probeX	Guide probe position in X
probeY	Guide probe position in Y
m2	M2 parameters
focuValue	
info	General information
telescop	Telescope name
version	TCS software version in use
oper	Telescope Operator name
site	Site information
longi	Longitude
lati	Latitude
level	Height above see level

Using the tifTCS_PUBLIC class

1. Configure the TCS database to allow scanning from the remote user environment.
   To this purpose it is necessary to add a scan device for the environment in the TCS database.
   For example, add the following lines in the USER.db file of the TCS environment:
   POINT "<VLT scan dev>" ccs_config:scan\ config:LAN:wsIns
   BEGIN
   ALIASwsIns
   END
2. Configure the user database to perform scanning from the TCS environment.
   To this purpose it is necessary to add a scan device for the TCS environment in the user database.
   For example, add the following lines in the USER.db file of the user (instrument) environment:
   POINT "<VLT scan dev>" ccs_config:scan\ config:LAN:wsTcs
   BEGIN
   ALIAS wsTcs
   END
3. Add an instance of the tifTCS_PUBLIC class anywhere in the user database. The Makefile must contain proper definitions for the TARGET_TCS to be built (see section 4.7.2)
4. Configure the scan system by running the tifTcsScanSetup command.