EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral

Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VLT PROGRAMME

ACCEPTED BY ESO Technical Archive 10.10.2006

VERY LARGE TELESCOPE

FINITO Tip Tilt Platform
Control Software Detailed Design

Doc. No.: VLT-SPE-ESO-15430-2786

Issue: 2

Date: 31/08/2006

Prepared:	Bertrand Bauvir, Henri Bonnet Name	31/08/2006 Date	Signature
Approved:	Anders Wallander, Markus Schöller Name	29/05/06 Date	Signature
Released:	Andreas Kaufer Name	29/05/06 Date	Signature

CHANGE RECORD

ISSUE	DATE	SECTION/PAGE AFFECTED	REASON/INITIATION DOCUMENTS/REMARKS
1/prep.1	11/08/2002	All	First draft
1/prep.2	16/10/2002	All	Second draft prepared for review
1/prep.3	14/01/2003	All	Third draft after review
1	25/01/2003	All	First release
2/prep.1	31/07/2006	2.1, 2.2 & 2.4	Add requirements related to IRIS Fast Guiding and Flux Tracking.
		4.1 & 4.2	Enhance TAC algorithm structure and add new commands.
2/prep.2	04/08/2006	4.1 & 4.2	Flux tracking stops when converged.
2/prep.3	06/08/2006	4.2	Modify CENTER command.
2/prep.4	07/08/2006	All	FTK renamed BTK.
2/prep.5	20/08/2006	4.3	Add dedicated point for configuration.
2	31/08/2006		Second release after review.

TABLE OF CONTENTS

TABLE OF FIGURES

This page has been left intentionally blank.

1 Introduction

1.1 Purpose

This document aims to present the design of FINITO Tip Tilt Platform Control Software that is part of FINITO Alignment and Compensation Unit. It is intended to be used as an input for the design of FINITO Workstation Software as well as a manual for user operation.

1.2 Scope

This document shall first shortly describe FINITO Alignment and Compensation Unit. It will then present the requirements and constraints associated to this software component. Some design decisions are then issued from these requirements.

The LCC libraries, messaging system and database are documented in [AD#02].

In order to ensure compliancy with LCU standard interface and behaviour described in [AD#01], the code of this software component was first generated from LCU Server Framework (LSF) template and extended following the guidelines documented in [AD#03].

The design description section of this document mainly focuses on the application-specific extensions brought to LSF and the TAC-based real-time computation required to provide the specified functionality.

In addition, the reader should be familiar with VLT software standards.

1.3 Applicable Documents

The following documents, of the issue shown if specified, form part of this manual to the extent specified herein. In the event of conflict between this document and those referenced, the content of this document shall be considered as a superseding requirement unless explicitly stated otherwise herein.

Reference	Document Number	Issue	Date	Title
[AD#01]	VLT-MAN-ESO-17210-0667	1.0	03/12/1996	Guidelines for the Development of VLT Application Software
[AD#02]	VLT-MAN-SBI-17210-0001	3.5	20/10/1999	LCU Common Software User Manual
[AD#03]	VLT-MAN-ESO-17210-2252	2	05/07/2002	LCU Server Framework User Manual
[AD#04]	VLT-MAN-ESO-17210-2970	2	21/03/2004	Tools for Advanced Control User Manual
[AD#05]	VLT-MAN-ESO-17210-3525	1	27/01/2005	RTDScope User Manual
[AD#06]	VLT-SPE-ESO-15420-2576	1	20/11/2001	FINITO Software Top Level Design Description
[AD#07]	VLT-SPE-ESO-15420-2575	1	17/07/2001	FINITO Software User Requirements
[AD#08]	VLT-SPE-ESO-15430-4021	1	31/08/2006	FINITO Beam Acquisition and Stabilization Technical Specification

1.4 Reference Documents

The following documents contain additional information that can be useful to the reader:

Reference	Document Number	Issue	Date	Title
[RD#01]	VLT-SPE-OAT-15430-0001	1.0	23/03/2001	FINITO Overall Description
[RD#02]	VLT-SPE-OAT-15430-0003	1.0	23/03/2001	FINITO Electronics Design
[RD#03]	N/A	01N	NOV 97	MPV955 Operating Manual
[RD#04]	VLT-SPE-ESO-15430-2790	1	31/08/2006	FINITO Detection Algorithm (As-built) Control Software Detailed Design

1.5 Acronyms

This document employs several abbreviations and acronyms to refer concisely to an item, after it has been introduced. The following list is aimed to help the reader in recalling the extended meaning of each short expression:

Acronym	Meaning
ACU	Alignment and Compensation Unit
AD	Applicable Document
ADC	Atmospheric Dispersion Compensation
ADU	Astronomical Detector Unit
BTK	Beam Tracking
CPU	Central Processing Unit
DD	Design Decision
LADC	Longitudinal Atmospheric Dispersion Compensation
LCU	Local Control Unit
LCC	LCU Common Software
LSF	LCU Server Framework
IFG	IRIS Fast Guiding
IO	Input/Output
OPD	Optical Path Difference
PPC	Power PC
RD	Reference Document
RMN	Reflective Memory Network
SR	Software Requirement
TAC	Tools for Advanced Control
TADC	Transversal Atmospheric Dispersion Compensation
TBC	To Be Confirmed
TBD	To Be Defined
TIM	Time Interface Module
TTP	Tip Tilt Platform
TST	Translation Stage
VLT	Very Large Telescope
VLTI	VLT Interferometer
ZOPD	Zero Optical Path Difference

2 Functional Specification

2.1 Overview

An overall description of FINITO can be found in [RD#01].

The Alignment and Compensation Unit (ACU) provides control over three active units, one for each FINITO beam. Each unit is constituted by a flat mirror mounted on a piezoelectric platform providing tip and tilt action. This platform rests on a motorized translation stage in a way that the reflected light beam is parallel to the translation axis when the piezoelectric platform is in its center position.

The ACU function is multiple:

- Initial alignment and optical path balancing of the telescope beams on FINITO.
- Optimization of the astronomical beam injection.
- Compensation of the atmospheric effects due to the observing geometry.
- Application of OPD offsets required by external instruments while fringe tracking.

These four functions are shortly presented below. A complete description can be found in [AD#06].

2.1.1 Optical Alignment

The first function of the ACU is the alignment of FINITO with respect to VLTI beams. The Tip Tilt Platform (TTP) is moved in order to maximize the photon flux on the corresponding Astronomical Detector Unit (ADU) photometric channel. The Translation Stage (TST) is moved to have fringes at internal Zero Optical Path Difference (ZOPD).

2.1.2 Astronomical Beam Injection

The second function of the ACU is the stabilization of the scientific beams on the FINITO input fiber; real-time stabilization of the scientific beams is required to minimize the effect of tip/tilt, induced by turbulence in the light-duct or tunnel, on FINITO operation. FINITO requires an external imager to provide it with fast tip/tilt error information.

2.1.3 Atmospheric Dispersion Compensation

The third function of the ACU is to perform Transversal and Longitudinal Atmospheric Dispersion Compensation (TADC and LADC, respectively). In general, the scientific instrument may operate at a different wavelength than FINITO. Therefore, a mismatch between the respective positions of ZOPD may arise due to the different optical path along the common air column, as well as a different apparent position in the sky associated to differential air refraction. An Atmospheric Dispersion Compensation (ADC) is required to correct these two atmospheric effects.

2.1.4 OPD Offsets while Fringe Tracking

When VLTI is operated in fringe tracking, the delay lines are controlled to maintain zero OPD condition among telescope beams as measured by FINITO ADU. An instrument may require VLTI to apply an OPD offset between the telescope beams in order, i.e., to center the scientific fringe packet on the instrument detector. Such offset is produced by applying the same offset with opposite sign between FINITO beams by means of the TST motion in order to drag the OPD, as seen by the instrument, to the requested position.

The purpose of this software is the control of FINITO TTP to provide the transversal optical alignment, optimal astronomical beam injection and TADC functions of the ACU. The computation of the TADC for the observed target is not part of the scope of this software.

The TADC requirements can be found in [AD#07] and are listed hereafter as a reminder.

Range	±1mrad
Resolution	< 0.1µrad
Update rate	< 1Hz

The TTP are Physik Instrumente S-330.10 Piezo Tip/Tilt operating in closed loop. They are delivered with high voltage amplifiers with -2V - 12V analog input signal.

2.2 Software Requirements

The main software requirements (SR) are extracted from [AD#06] and [AD#08]:

Reference	Requirement
[SR#1]	Provide angular control over the FINITO TTP platforms.
[SR#2]	Provide an automatic procedure to align FINITO with respect to the scientific beams.
[SR#3]	Provide a mechanism to drive the TTP from the error vector measured by the VLTI laboratory guider (i.e. IRIS).

2.3 Constraints and Performance

The main constraints and performance requirements associated with this software are:

Reference	Dependency	Constraint
[Constr#1]	N/A	Provide a VLT standard interface to the outside world (Command interpreter, Online database Access, Error reporting, etc.).
[Constr#2]	[SR#3]	Minimize the delay between the measurement and application of the guiding error vector measured by IRIS.

2.4 Design Decisions

The following list of design decisions (DD) is issued a priori from the above requirements:

Reference	Dependency	Decision
[DD#1]	[Constr#1]	The software will be based on LSF.
[DD#2]	[SR#1]	The TTP angles will be remotely controlled by means of high resolution analog signals.
[DD#3]	[DD#2]	As six analog channels are required, the analog interface will be provided by a Pentland Systems MPV955 DAC board.
[DD#4]	[DD#3]	In order to reuse the coding effort conducted for the design of FINITO Fiber Modulator Unit, the software will be based on TAC[1].
[DD#5]	[SR#2]	Alignment of FINITO with respect to the input beams will use a synchronous demodulation mechanism.
[DD#6]	[DD#5]	Alignment of FINITO with respect to the input beams will receive the associated photometric channel flux from the ADU by means of the VLTI Reflective Memory Network (RMN).
[DD#7]	[SR#3]	IRIS error vector is passed to the TTP by means of the VLTI RMN.
[DD#8]	[Constr#2]	IRIS activity is not synchronized to the TIME bus. The TTP real-time activity will be triggered by the TIM board and executed at 2kHz in order to minimize communication delay.

[1] At the time of this design, no VLT compliant VxWorks driver was available and LSF did only provide interface to the VLT standard VMIC VMIVME3111 analog IO board.

3 Architecture Overview

3.1 Hardware

The following LCU architecture has been selected for this application:

- 1 Motorola PPC CPU board.
- 1 VMIC VMIVME5576 Reflective Memory board.
- 1 ESO Time Interface Module (TIM) board.
- 1 Pentland Systems MPV955 16 bits resolution Digital to Analog Converter (DAC) board.

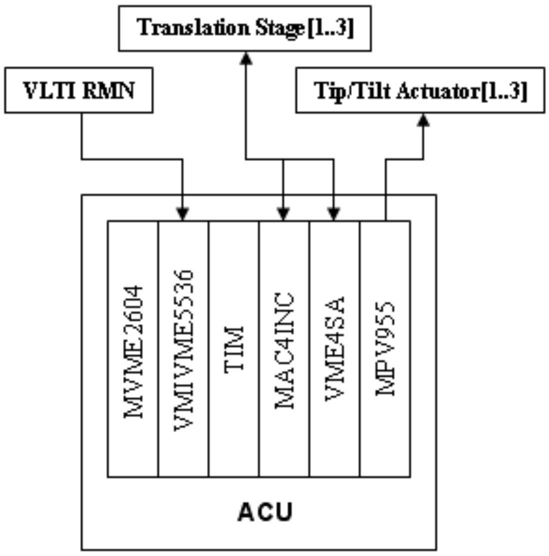

Figure 1: ACU LCU Hardware structure

The other boards (i.e. motor controller board and amplifier) present in the ACU LCU are dedicated to the control of FINITO TST.

The design assumes the following analog signal interface.

Signal	MPV955	Range
TXTP0 axis control	Channel#0	±10V
TYTP0 axis control	Channel#1	±10V
TXTP1 axis control	Channel#2	±10V
TYTP1 axis control	Channel#3	±10V
TXTP2 axis control	Channel#4	±10V
TYTP2 axis control	Channel#5	±10V

3.2 Software

3.2.1 Application Package

The software is responsible for the control of the TTP subsystem of FINITO. It encompasses the following components:

- A command interface, based on CCS message system and on a command interpreter running on the CPU board of the ACU LCU.
- A public database providing status and signals information.
- An engineering graphical user interface for test or maintenance purposes.

Compliancy with [AD#01] is ensured by the use of LSF to implement standard behaviour and to generate part of the application-specific code (command handling routines) and database structure. The application software was generated using *lsfConfig* tool, LSF configuration file can be found in the appendix.

The application software architecture is described in [AD#06]. Interaction with the TTP is done through an instance of a LSF software device. The real-time control algorithm is exploiting TAC application and its library of function blocks.

3.2.2 Support Packages

The application makes extensive use of the following support packages to implement the standard LCU behaviour and TTP control algorithm and provide the user with an engineering graphical interface.

Package Description	Name
LCU Common Software	lcc
LCU Server Framework	lsf
Tools for Advanced Control	tac
RMN Data Interface	rmac
Real-Time Plotting Tools	rtdscope

3.2.3 CMM Modules

This application encompasses the following CMM modules:

fntttp

This module contains the complete set of files that compose this application software:

- Application-specific software code.
- Online database classes files.
- Online database configuration files.
- Command definition and interface tables (CDT and CIT files).
- TAC algorithm configuration file.
- Engineering graphical user interface panels.

VLT-SPE-ESO-15430-2786

This module shall be used to document the historical development of this document. At least all drafts, preparations and releases that are referenced in any other documents shall be archived as an own version within this module.

4 Design Description

This section presents the design of the fntttp application software.

4.1 TAC Control Algorithm

The following diagram represents the TAC algorithm that implements the control of FINITO TTP angular positions. The blocks appearing in blue are instantiated three times, once for each beam.

In order to avoid rounding errors, <*thetaX*> and <*thetaY*> are expressed in mrad at the level of TAC algorithm, while external interface is defined in radians.

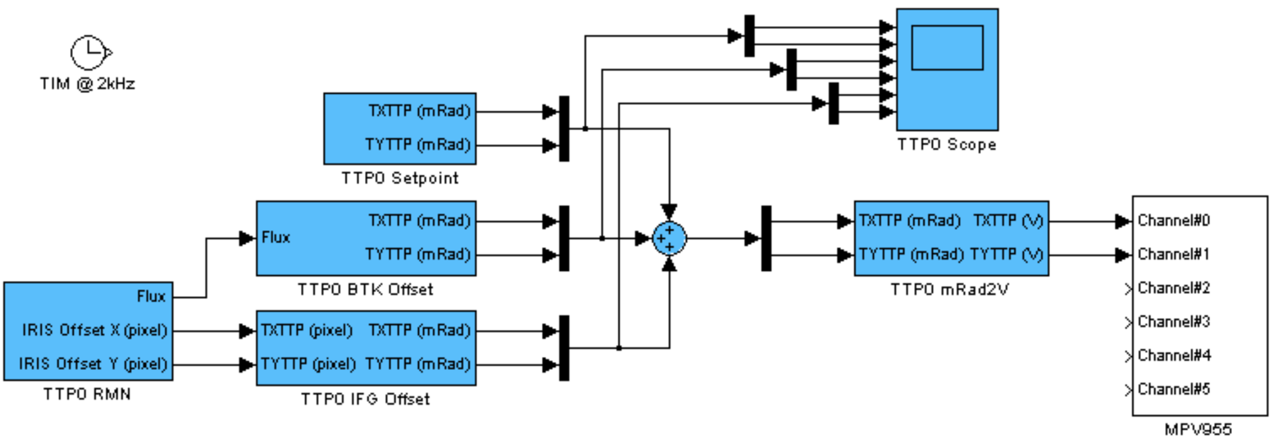

Figure 2: TAC Algorithm schematic (one beam)

It encompasses the following parts:

- A block to configure the TIM board to generate interrupts at 2kHz rate.
- A block to access the required data on the VMIVME5536 RMN board.
- A block to access the Pentland Systems MPV955 DAC board.
- A data monitoring block to allow real-time inspection (using rtdscope application) or recording in a file the different control signals (using TAC record mechanism).
- A set of blocks that provides the user with an interface to define the TTP absolute angular position.
- A Beam Tracking (BTK) function to center the scientific beams on the input optical fiber (i.e. maximize the flux injected into FINITO).
- A Fast Guiding (IFG) function to minimize the beam wander induced by light-duct turbulence.
- A function to convert the TTP angular position from laboratory to equipment reference system.

4.1.1 Absolute Setpoint

Two constant blocks provide the user with an interface to define the TTP absolute angular position (i.e. low accuracy alignment, TADC offset, etc.).

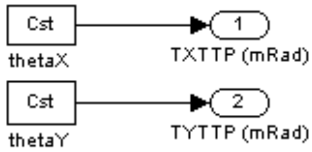

Figure 3: External setpoint

4.1.2 Beam Tracking

Accurate centering of the scientific beams onto the FINITO input fibers is achieved by means of synchronous demodulation. The principle is to apply a harmonic circular modulation to the TTP; the flux measured on the associated photometric channel will feature a component at the modulation frequency if the beam is not centered onto the input fiber, the phase of which defines the direction of the de-centering.

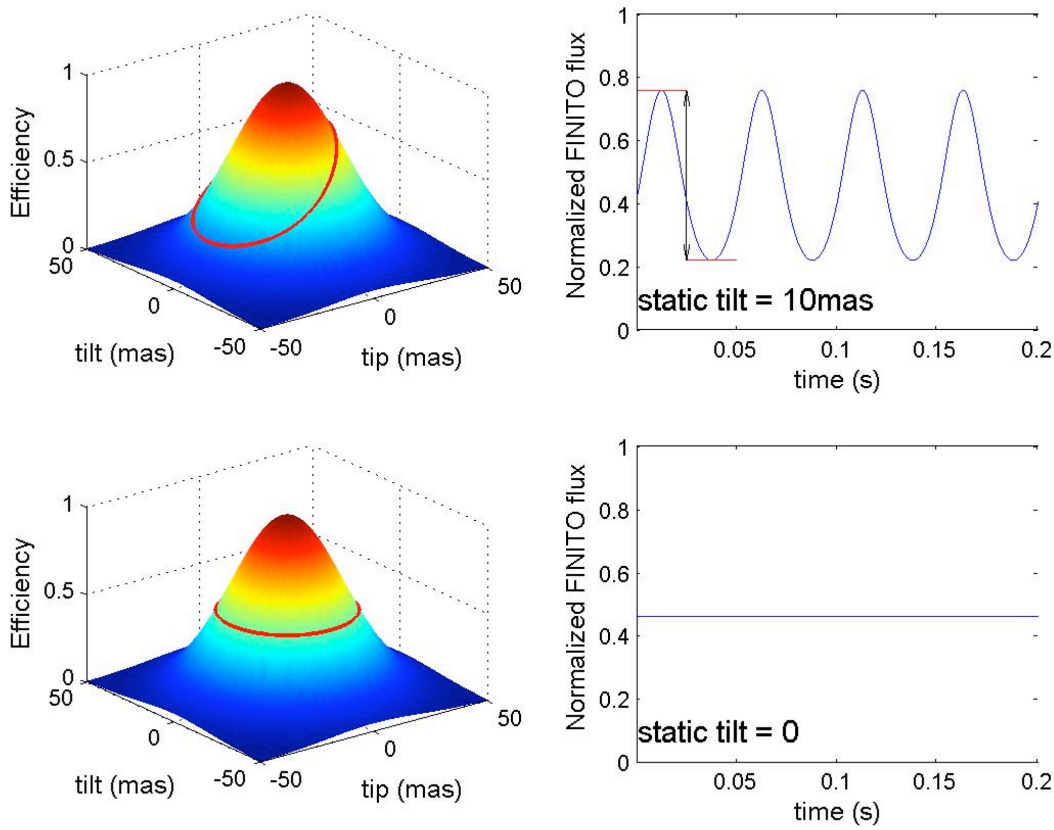

Figure 4: Principle of synchronous demodulation

Automatic centering is achieved by, e.g., multiplying the normalized ADU flux with the modulation signal, extracting its DC component (the sign of which defines in which direction is the center of the fiber and the amplitude of which is proportional to the centering error) and integrating it to converge to the center of the fiber.

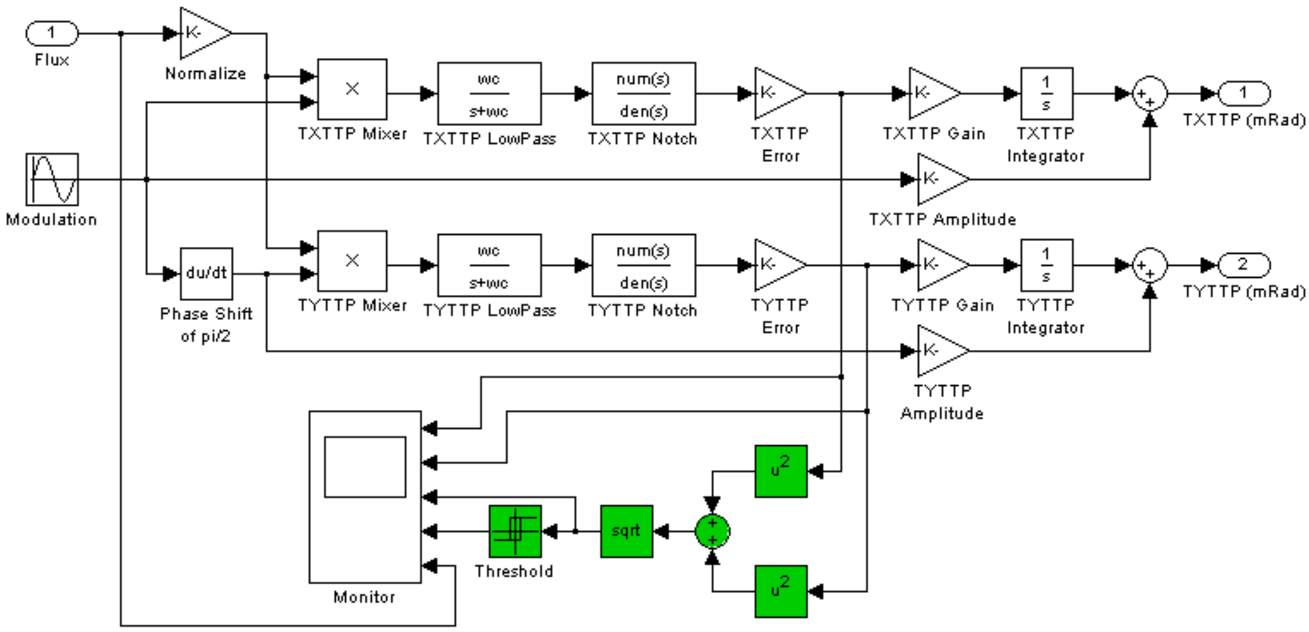

Figure 5: Alignment using synchronous demodulation

The notch filters are used to reject further (i.e. further than the rejection provided by the low-pass filters) the circular modulation signal from the integrator input.

The centering error is estimated from the individual tip/tilt error signals and passed into a comparator function.

The signal monitoring part is used to report to the application server the flux and state (converged or not) of the centering loop. The server is then capable of measuring at a slow rate the average flux (for flux normalization purpose) and automatically stopping modulation automatically when the estimated centering error is sufficiently small; the function appearing in green on the previous schematic may be executed in the data monitoring callback as it is not essential to the real-time control of the TTP.

The configurable parameters are the modulation <*amplitude*>, integrator <*gain*> and comparator <*threshold*>. The frequency is not configurable a priori but shall be selected within vibration-free frequency regions.

4.1.3 Fast Guiding

An open-loop guiding strategy has been elected. The scientific beams are aligned with respect to VLTI reference by means of slow guiding the telescope probe from IRIS imager located in the laboratory; FINITO ACU applies a TTP position offset from each measured error vector[2].

[2] E.g. the slow guiding of the telescope probe will not respond to a 1Hz displacement of the beam around IRIS guiding pixel but FINITO will still require a local compensation to maintain optimal alignment.

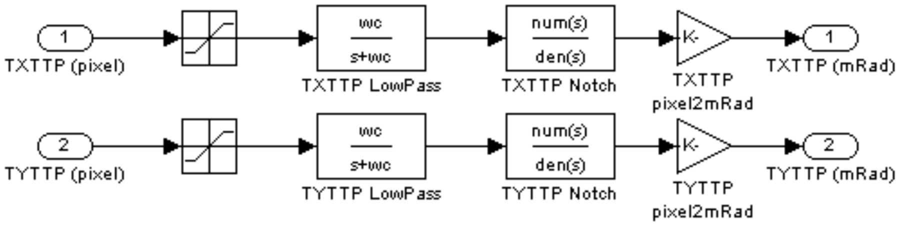

Figure 6: IRIS fast guiding

IRIS error vector is first saturated to avoid driving the TTP during beam acquisition, filtered to avoid amplification of high frequencies due to delay and converted from pixel scale into mrad.

The configurable parameters are the saturation <*thresholds*> and <*pixel2mRad*> conversion factors.

4.1.4 Conversion into the Equipment Reference System

Each TTP may be misaligned with respect to the vertical and horizontal axes. Therefore, the angular positions are first passed through a rotation matrix to generate the requested angular position around the TTP physical axes. The associated voltage is computed using a linear law and then fed to the MPV955 DAC board.

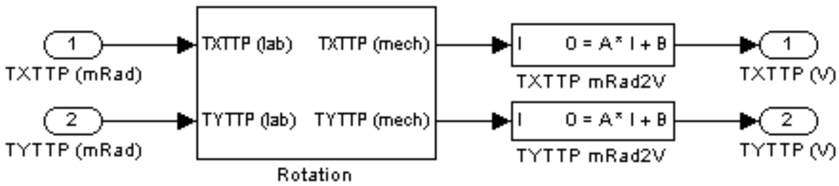

Figure 7: Conversion into equipment characteristics

The configurable parameters are the rotation <*angle*> (physical axes are considered orthogonal) and conversion factors <*slope*> and <*offset*>.

4.2 Commands

4.2.1 Standard Commands (with application-specific extended behaviour)

Command	Description
`STOP`	Stops ongoing modulation and beam centering procedures.

4.2.2 Public Commands

Command	Parameters	Reply	Description
`SETTILT`	<*ttpId*> <*thetaX*> <*thetaY*>	Ok Error	Requests the specified TTP to move to a new angular position. <*thetaX*> and <*thetaY*> are expressed in rad. This command modifies the parameters of the setpoint constant blocks. The command returns when the motion is complete or in case of failure.
`GETTILT`	<*ttpId*>	<*thetaX*> <*thetaY*>	Retrieves the current position in rad of the specified TTP.

`CENTER`	*<ttpId>*	Ok Error	Requests the specified TTP to move to the center of the angular range.
`STRTBTK`	*<ttpId>*	Ok Error	Requests the specified TTP to center the beam onto the input fiber. The command returns when the motion is complete or in case of failure. The following procedure is applied: • Starts the circular modulation with the specified <*amplitude*>. • Closes the beam centering loop by setting the integrator <*gain*> to its nominal value. • Attaches a callback to the associated data Monitor block to measure the average flux (periodically update the flux normalization gain) and the estimated centering error. • When the centering error goes below a threshold or has not yet reached the threshold after a configurable time, the centering loop is disabled by setting the modulation <*amplitude*> and integrator <*gain*> to zero.
`STOPBTK`	*<ttpId>*	Ok Error	Stops an ongoing beam centering procedure.

4.2.3 Maintenance Commands

Command	Parameters	Reply	Description
`ENAIFG`	*<ttpId>*	Ok Error	Enables IFG (i.e. the fast control of the TTP from the guider error vector) for the specified TTP. This is achieved by setting the <*pixel2mRad*> gain parameter to its nominal value.
`DISIFG`	*<ttpId>*	Ok Error	Disables IFG for the specified TTP. This is achieved by setting the <*pixel2mRad*> gain parameter to zero.
`GETIFG`	*<ttpId>*	*<status>*	Returns the status (boolean) of IFG function.
`ENABTK`	*<ttpId>*	Ok Error	Enables BTK (i.e. automatic beam centering) for the specified TTP. This is achieved by starting the circular modulation setting the modulation <*amplitude*> parameter and integrator <*gain*> to their nominal value.
`DISBTK`	*<ttpId>*	Ok Error	Disables BTK for the specified TTP. This is achieved by setting the <*amplitude*> and integrator <*gain*> parameters to zero.
`GETBTK`	*<ttpId>*	*<status>*	Returns the status (boolean) of BTK function.

4.2.4 Test Commands

Command	Parameters	Reply	Description
`ENAMOD`	*<ttpId>*	Ok Error	Enables the circular modulation for the specified TTP. This is achieved by starting the circular modulation setting the modulation <*amplitude*> parameter. The integrator <*gain*> parameter is maintained to zero.
`DISMOD`	*<ttpId>*	Ok Error	Disables the circular modulation for the specified TTP. This is achieved by setting the <*amplitude*> parameter to zero.
`GETMOD`	*<ttpId>*	*<status>*	Returns the status (boolean) of modulation function.

4.3 Online Database

The online database structure is directly inherited from LSF with the extensions described below.

4.3.1 Control Branch

No extension to the LSF control branch is foreseen in the scope of this application software.

4.3.2 Configuration Branch

The configuration branch will contain the following application-specific configuration attributes for each TTP0..2 point:

Point	Attribute	Type	Description
`:ifg`	`.pixel2rad`	Double	IFG conversion factor from IRIS pixel scale to radian.
`:ifg`	`.saturation`	Double	IFG saturation threshold (in pixel scale).
`:ifg`	`.enable`	Boolean	Set to operate the TTP in IFG by default.
`:btk`	`.gain`	Double	BTK integrator <*gain*>. The same parameter is applied to both axes of the TTP.
`:btk`	`.amplitude`	Double	BTK modulation <*amplitude*>. The same parameter is applied to both axes of the TTP.
`:btk`	`.threshold`	Double	BTK convergence criterion.
`:btk`	`.timeout`	Double	Time after which `STRTBTK` will return an error if not yet converged.

The other parameters required for the real-time control algorithm are not dynamically modified by the software and therefore appear in the TAC algorithm configuration file.

4.3.3 Data Branch

The data branch will at least gather the following attributes for each TTP0..2 point:

Point	Attribute	Type	Description
	`.thetaX`	Double	Current absolute position in rad of the TTP X axis.
	`.thetaY`	Double	Current absolute position in rad of the TTP Y axis.
`:ifg`	`.thetaX`	Double	Current IFG offset (averaged over 1s) in rad for the TTP X axis.
`:ifg`	`.thetaY`	Double	Current IFG offset (averaged over 1s) in rad for the TTP Y axis.
`:ifg`	`.status`	Integer	IFG status (0 – undefined, 1 – off and 2 – on).
`:btk`	`.thetaX`	Double	Current BTK offset (averaged over 1s) in rad for the TTP X axis.
`:btk`	`.thetaY`	Double	Current BTK offset (averaged over 1s) in rad for the TTP Y axis.
`:btk`	`.error`	Double	Estimated BTK centering error (averaged over 1s) in rad.
`:btk`	`.status`	Integer	BTK status (0 – undefined, 1 – off and 2 – on).

5 User Manual

5.1 Installing the Software

The steps to install the fntttp application software on a LCU and workstation environment may be extracted from in the environment templates of the software test suite.

```
prompt > cmmCopy fntttp
prompt > find fntttp -name src | xargs -I DIR \
> sh -c "cd DIR && make all install"
prompt > cd fntttp/lcu/test/ENVIRONMENTS
prompt > export WSENV=<wsEnv>
prompt > export LCUENV=<lcuEnv>
prompt > vccEnvCreate –e $WSENV –s ./wsTat
prompt > vccEnvCreate –e $LCUENV –t LCU –h $HOST –d /vltdata/ENVIRONMENTS/$LCUENV \
> -s ./lcuTat –w $WSENV
prompt > vccEnvStart –e $WSENV
prompt > lccBoot $LCUENV
```

5.2 Real-time Display and Engineering Graphical User Interface

The fntttp application provides an engineering graphical user interface and a real-time display panel exploiting rtdscope application. The two panels are started with the following shell commands:

```
prompt > fntttpgui &
prompt > rtdscopeGui &
```

The update of the real-time display is started putting the two applications ONLINE:

```
prompt > msgSend $LCUENV fntttpServer ONLINE ""
prompt > msgSend $LCUENV rtdscopeServer ONLINE ""
```

The real-time display shows by default TTP0 control signals. Change the TTP the real-time display is connected to by:

```
prompt > msgSend $LCUENV rtdscopeServer STANDBY ""
prompt > dbWrite "@$LCUENV:rtdscope:control:misccfg.block" "TTP2_RTD_Scope"
prompt > msgSend $LCUENV rtdscopeServer ONLINE ""
```

5.3 Configuring the Software

5.3.1 Conversion to Equipment Reference System

The software is configured assuming that ±10V on the output of the MPV955 board corresponds to ±1mrad. The conversion parameters are part of the TAC configuration file and may be changed to fit the actual hardware reality. The procedure assumes that an artificial light source is fed to FINITO and that the angular displacement of the beam axis is monitored using a sighting telescope.

The translation blocks apply the standard linear formula V = slope.mrad + offset. The following example concerns TTP0.

The best way to determine the actual slope and offset parameters is first to tune the voltage offset.

Center the TTP (at software level) with the following commands:

```
prompt > msgSend $LCUENV fntttpServer ONLINE ""
prompt > msgSend $LCUENV fntttpServer CENTER 0
```

Then change the offset with the following commands until the TTP is properly centered on the X and Y axes.

```
prompt > msgSend $LCUENV tacServer MODBLCK TTP0_Convert_TX_mrad2V 0.0,<offsetX>
prompt > msgSend $LCUENV tacServer MODBLCK TTP0_Convert_TY_mrad2V 0.0,<offsetY>
```

Measure the orientation of the TTP physical axes with respect to the horizontal and vertical references. Fix the offset parameters and send the TTP X axis to 1mrad (at software level) with the following command:

```
prompt > msgSend $LCUENV fntttpServer SETTILT 0,0.001,0.0
```

Modify the rotation matrix angle in order for the light spot to move along the horizontal reference axis:

```
prompt > msgSend $LCUENV tacServer MODBLCK TTP0_Convert_Rotation <angle>
```

Change the slope with the following command until the TTP is properly deviating the incoming beam by 1e-3 rad.

```
prompt > msgSend $LCUENV tacServer MODBLCK TTP0_Convert_TX_mrad2V <slopeX>,<offsetX>
```

Send the TTP Y axis to 1mrad (at software level):

```
prompt > msgSend $LCUENV fntttpServer SETTILT 0,0.0,0.001
```

Verify that the rotation angle is correct (the light spot should have moved along the vertical reference axis). Repeat for the Y axis:

```
prompt > msgSend $LCUENV tacServer MODBLCK TTP0_Convert_TY_mrad2V <slopeY>,<offsetY>
```

The new parameters may be saved by generating a new TAC configuration file:

```
prompt > msgSend $LCUENV tacServer REPORT fntttp.tac
```

It is essential that the measured parameters be archived in the fntttp CMM module.

6 Appendix

6.1 CDT and CIT files

6.1.1 fntttpPublic.cdt

```
//*************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: fntttpPublic.cdt,v 1.29 2006/06/28 17:21:01 vltsccm Exp $"
//
// who       when        what
// --------  ----------  ----------------------------------------------
// bbauvir   2001-10-25  created
//------------------------------------------------------------------------

//========================================================================
// Specific
//========================================================================

COMMAND=  SETTILT
SYNONYMS=  fntttpSoftDevSetTiltPosition
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

    PAR_NAME= thetaX
    PAR_TYPE= REAL
    PAR_RANGE= INTERVAL MIN=-0.001;MAX=0.001
    PAR_DEF_VAL=0

    PAR_NAME= thetaY
    PAR_TYPE= REAL
    PAR_RANGE= INTERVAL MIN=-0.001;MAX=0.001
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Set Tip Tilt Platform angles (rad).
@

COMMAND=  GETTILT
SYNONYMS=  fntttpSoftDevGetTiltPosition
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
REPLY_PARAMETERS=
    PAR_NAME= thetaX
    PAR_TYPE= REAL

    PAR_NAME= thetaY
    PAR_TYPE= REAL

HELP_TEXT=
```

```
Retrieves Tip Tilt Platform angles (rad).
@

COMMAND=  CENTER
SYNONYMS=  fntttpSoftDevCenterPosition
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Center the TTP to the middle of the angular range.
@

COMMAND=  STRTBTK
SYNONYMS=  fntttpSoftDevStartBeamTracking
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Start beam tracking function. The function returns when the beam is
acquired. It is interrupted by STOPBTK command.
@

COMMAND=  STOPBTK
SYNONYMS=  fntttpSoftDevStopBeamTracking
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Stop beam tracking function.
@

//
//========================================================================
//
// ___oOo___
//
```

6.1.2 fntttpMaintenance.cdt

```
//*************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: fntttpMaintenance.cdt,v 1.29 2006/06/28 17:21:01 vltsccm Exp $"
//
// who       when        what
// --------  ----------  ----------------------------------------------
// bbauvir   2001-10-25  created
//------------------------------------------------------------------------

//========================================================================
// Specific
//========================================================================

COMMAND=  ENAIFG
SYNONYMS=  fntttpSoftDevEnableIRISGuiding
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Enable IRIS guiding for the specified Tip Tilt Platform.
@

COMMAND=  DISIFG
SYNONYMS=  fntttpSoftDevDisableIRISGuiding
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Disable IRIS guiding for the specified Tip Tilt Platform.
@

COMMAND=  GETIFG
SYNONYMS=  fntttpSoftDevGetIRISGuiding
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0
REPLY_FORMAT = A
HELP_TEXT=
Retrieves Tip Tilt Platform IFG status (bool).
@

COMMAND=  ENABTK
SYNONYMS=  fntttpSoftDevEnableBeamTracking
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0
```

```
REPLY_FORMAT = A
HELP_TEXT=
Enable Beam Tracking for the specified Tip Tilt Platform.
@

COMMAND=  DISBTK
SYNONYMS=  fntttpSoftDevDisableBeamTracking
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Disable Beam Tracking for the specified Tip Tilt Platform.
@

COMMAND=  GETBTK
SYNONYMS=  fntttpSoftDevGetBeamTracking
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0
REPLY_FORMAT = A
HELP_TEXT=
Retrieves Tip Tilt Platform Beam Tracking status (bool).
@

//
//==========================================================================
//
// ___oOo___
//
```

6.1.3 fntttpTest.cdt

```
//**************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: fntttpTest.cdt,v 1.29 2006/06/28 17:21:01 vltsccm Exp $"
//
// who       when        what
// --------  ----------  ------------------------------------------------
// bbauvir   2001-10-25  created
//--------------------------------------------------------------------------

//==========================================================================
// Specific
//==========================================================================

COMMAND=  ENAMOD
SYNONYMS=  fntttpSoftDevEnableModulation
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
```

```
HELP_TEXT=
Enable circular modulation for the specified Tip Tilt Platform.
@

COMMAND=  DISMOD
SYNONYMS=  fntttpSoftDevDisableModulation
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0

REPLY_FORMAT = A
HELP_TEXT=
Disable circular modulation for the specified Tip Tilt Platform.
@

COMMAND=  GETMOD
SYNONYMS=  fntttpSoftDevGetModulation
FORMAT= A
PARAMETERS=
    PAR_NAME= platformId
    PAR_TYPE= INTEGER
    PAR_RANGE= INTERVAL MIN=0;MAX=2
    PAR_DEF_VAL=0
REPLY_FORMAT = A
HELP_TEXT=
Retrieves Tip Tilt Platform modulation status (bool).
@

//
//=========================================================================
//
// ___oOo___
//
```

6.1.4 fntttpSoftDev.cit

```
//*************************************************************************
// E.S.O. - VLT project
//
// "@(#) $Id: fntttpSoftDev.cit,v 1.29 2006/06/28 17:21:01 vltsccm Exp $"
//
// who       when        what
// --------  ----------  -------------------------------------------------
// pduhoux   2000-07-04  created
//-------------------------------------------------------------------------
//
// This file has been generated by a utility
//
//  !!!!!!!!!!!  DO NOT MANUALLY EDIT THIS FILE  !!!!!!!!!!!
//
//*************************************************************************

//=========================================================================
// Specific
//=========================================================================
SETTILT, IfntttpSoftDevSetTiltPosition,   FUNCTION
GETTILT, IfntttpSoftDevGetTiltPosition,   FUNCTION
CENTER,  IfntttpSoftDevCenterPosition,    FUNCTION
STRTBTK, IfntttpSoftDevStartBeamTracking, TASK,fntttpBTK,,,30000
STOPBTK, IfntttpSoftDevStopBeamTracking,  FUNCTION
ENAIFG,  IfntttpSoftDevSetIRISGuiding,    FUNCTION
DISIFG,  IfntttpSoftDevSetIRISGuiding,    FUNCTION
```

```
GETIFG,  IfntttpSoftDevGetIRISGuiding,    FUNCTION
ENABTK,  IfntttpSoftDevSetBeamTracking,   FUNCTION
DISBTK,  IfntttpSoftDevSetBeamTracking,   FUNCTION
GETBTK,  IfntttpSoftDevGetBeamTracking,   FUNCTION
ENAMOD,  IfntttpSoftDevSetModulation,     FUNCTION
DISMOD,  IfntttpSoftDevSetModulation,     FUNCTION
GETMOD,  IfntttpSoftDevGetModulation,     FUNCTION
//
//===========================================================================
//
// ___oOo___
//
```

6.2 Online Database Description files

6.2.1 fntttpDB_DATA.class

```
// ************************************************************************
// * E.S.O. - VLT project
// *
// * "@(#) $Id: fntttpDB_DATA.class,v 1.29 2006/06/28 17:21:01 vltsccm Exp $"
// *
// * who       when        what
// * --------  ----------  ----------------------------------------------
// * pduhoux   2000-04-06  created
// *

#include "lsfDB_DATA.class"
#include "tacDB_DATA.class"

CLASS "BASE_CLASS" "fntttpDB_DATA_IFG"
BEGIN
    ATTRIBUTE int32  status 0
    ATTRIBUTE double thetaX 0
    ATTRIBUTE double thetaY 0
END

CLASS "BASE_CLASS" "fntttpDB_DATA_BTK"
BEGIN
    ATTRIBUTE int32  status 0
    ATTRIBUTE double thetaX 0
    ATTRIBUTE double thetaY 0
    ATTRIBUTE double error  0
END

CLASS "BASE_CLASS" "fntttpDB_DATA_TTP"
BEGIN
    ATTRIBUTE double thetaX 0
    ATTRIBUTE double thetaY 0
    ATTRIBUTE fntttpDB_DATA_IFG ifg
    ATTRIBUTE fntttpDB_DATA_BTK btk
#ifdef MAKE_VXWORKS
    ATTRIBUTE tacDB_MONIT_SIGNAL monitor
    BEGIN
        ATTRIBUTE Vector name(tacMAX_DATA_NUMBER,bytes64)
        BEGIN
            Value("TXTTP Setpoint (mRad)",
                  "TYTTP Setpoint (mRad)",
                  "TXTTP BTK offset (mRad)",
                  "TYTTP BTK offset (mRad)",
                  "Average flux (ADU)",
                  "TXTTP BTK error (mRad)",
                  "TYTTP BTK error (mRad)",
                  "","","")
        END
        ATTRIBUTE Vector signal(tacMAX_DATA_NUMBER,double)
        BEGIN
            Value(0,0,0,0,0,0,0,0,0,0)
        END
    END
#endif
END

CLASS "lsfDB_DATA" "fntttpDB_DATA"
BEGIN
    ATTRIBUTE fntttpDB_DATA_TTP TTP0
    ATTRIBUTE fntttpDB_DATA_TTP TTP1
```

```
    ATTRIBUTE fntttpDB_DATA_TTP TTP2
END

// END OF FILE
// ===============================================================
```

6.2.2 fntttpDB_CONFIG.class

```
// ***********************************************************************
// * E.S.O. - VLT project
// *
// *"@(#) $Id: fntttpDB_CONFIG.class,v 1.29 2006/06/28 17:21:01 vltsccm Exp $"
// *
// * who       when        what
// * --------  ----------  ----------------------------------------------
// * pduhoux   2000-04-06  created
// *

#include "lsfDB_CONFIG.class"

CLASS "BASE_CLASS" "fntttpDB_CFG_IFG"
BEGIN
    ATTRIBUTE double pixel2rad 0
    ATTRIBUTE double saturation 0
    ATTRIBUTE int32  enable 1
END

CLASS "BASE_CLASS" "fntttpDB_CFG_BTK"
BEGIN
    ATTRIBUTE double gain      0
    ATTRIBUTE double amplitude 0
    ATTRIBUTE double threshold 0
    ATTRIBUTE double timeout   0
END

CLASS "BASE_CLASS" "fntttpDB_CFG_TTP"
BEGIN
    ATTRIBUTE fntttpDB_CFG_IFG ifg
    ATTRIBUTE fntttpDB_CFG_BTK btk
END

CLASS "lsfDB_CONFIG" "fntttpDB_CONFIG"
BEGIN
    ATTRIBUTE fntttpDB_CFG_TTP TTP0
    ATTRIBUTE fntttpDB_CFG_TTP TTP1
    ATTRIBUTE fntttpDB_CFG_TTP TTP2
END

// END OF FILE
// ===============================================================
```

___oOo___