

VST Project

Telescope Control Software – Electronic Cabinet Cooling System VSTCOOL Module Description

Doc.no. : VST-SPE-ESO-25000-1082

Date: September 23, 2004
Issue: 1.0

*DRAFT VERSION. NOT VALID FOR REVIEW
FOR TECHNICAL DISCUSSION ONLY*

	Name	Date	Signature
<i>Prepared by</i>	P. Schipani, M. Brescia		
<i>Approved by</i>	D.Mancini		
<i>Released by</i>	D.Mancini		

Change Record

Issue/Rev.	Date	Section/Page affected	Reason/Remarks
1.0	23/09/04	All	First Issue

Index

1	Introduction	4
2	Purpose	4
3	Scope	4
4	Reference Documents	4
5	Description.....	5
6	Command Definition	7
6.1	Command Definition Table	7
6.2	Command Interpreter Table	14
7	Database Configuration	15
7.1	vstcool.class	15
7.2	vstcoolAZ.db	16
7.3	vstcoolALT.db	17
7.4	vstcoolADC.db	18
7.5	vstcoolM12.db	19
8	Configuration files	20
9	Abbreviation and Acronyms	21
10	Pendings.....	21

1 Introduction

The electronic cabinets of the VST telescope are cooled by cabinet thermal controllers provided by ESO. They are stand-alone controllers, which can be configured manually from their front panel and perform automatically the temperature control simply turning them on. Therefore no software control is foreseen for cabinet temperatures. The relevant parameters are collected by the *vstcool* software module and stored in the database, from where they are available to the user for monitoring and maintenance purposes.

2 Purpose

This purpose of this document is to describe the principles of operation of the *vstcool* module, its interface with the electronic cabinet thermal controllers provided by ESO and used to cool the VST cabinets, and the exchanged data.

3 Scope

It is assumed the reader has a good knowledge of the VLT programming standards.

4 Reference Documents

id.	Document code	Type	Source	Date	Issue
1	VLT-MAN-ESO-17130-1603	VLT Electronic Cabinet Cooling System User Manual	ESO	27.04.1998	1.0

5 Description

The *vstcool* module is installed in all the LCU environments controlling at least one ESO Cabinet Cooling Controller. The communication between the LCU and each controller is implemented by a RS232 serial line, through one channel of the ESD ISER8 serial communication board.

The correspondence is as follows:

ENVIRONMENT	1 ST CONTROLLER	2 ND CONTROLLER	3 RD CONTROLLER
ltvaz	Azimuth LCU Cabinet	Azimuth Drive Cabinet	Azimuth Power Cabinet
ltvalt	Altitude LCU Cabinet	Altitude Drive Cabinet	Altitude Power Cabinet
ltvadc	Ad/Rot LCU Cabinet		
ltvm12	M1-M2 LCU Cabinet	Center-Piece Cabinets no. 8-9-10	
ltvhb	HBS LCU Cabinet		

Tab. 1 – LCU / Cooling controllers correspondence

The application running on all the LCUs is the same. After the reboot the *vstcoolServer* goes automatically to ONLINE state. The user has normally no need to send commands to the *vstcoolServer* application.

The *vstcoolServer* is designed to communicate by default with three controllers. When the *vstcoolServer* is loaded during the LCU reboot, three background tasks start to communicate with three controllers through three different /iserX devices, which are configurable in the db ESOCOOL.DATAxch attributes .

If one of the ESOCOOL.DATAxch attribute in the db is not set to any /iserY channel, no communication is attempted for that channel and the task kills itself. Also, if no reply is received on any of the three channels (a normal situation in *ltvadc*, *ltvm12*, *ltvhb* environments, handling just one or a maximum of two controllers) the corresponding task kills itself. Otherwise if the communication between the task and the cabinet thermal controller is ok, some monitor information are periodically retrieved as far as the module is in ONLINE state.

The retrieved parameters are listed in the *vstcool.class* file. They are available in the workstation environment *wvtcs* through the scan system and can be monitored through the *vstcoolgui* panel (Fig. 1). A description of the parameters is available in Tab. 2.

DB Name	Description
CabinetTemp	Cabinet temperature
AmbientTemp	Ambient temperature
T_Inlet	Inlet temperature
T_Outlet	Outlet temperature
Flow1	Flow rate 1 (scaled flow rate measured by the 1 st transducer)
Flow2	Flow rate 2 (scaled flow rate measured by the 2 nd transducer)
Flow3	Flow rate 3 (scaled flow rate measured by the 3 RD transducer)
Cabinet_high	Cabinet temperature exceeds the maximum allowed
Outlet_high	Outlet temperature exceeds maximum (with reference to ambient temperature)
Inlet_high	Inlet temperature too high for required set-point
Airflow_insuff	Insufficient airflow
Coolant_insuff	Insufficient coolant flow

Tab. 2 – Parameters Description

An alarm is generated in the Alarm System if the cabinet temperature becomes too high (the threshold can be set in the *vstcool.class* file).

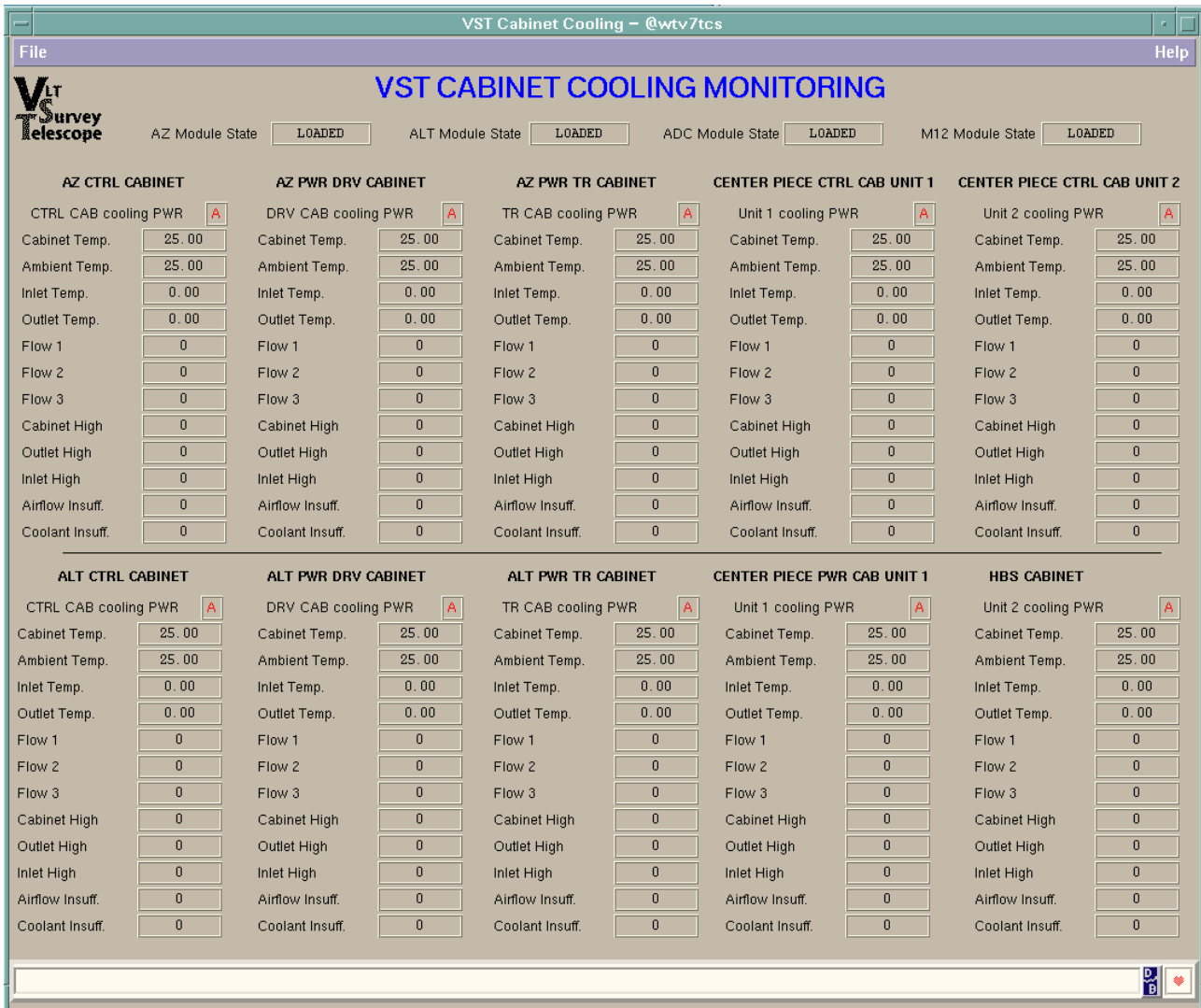


Fig. 1 – vstcoolgui panel

6 Command Definition

6.1 Command Definition Table

```
//*****
// O.A.C. - VST project
// #
// # "@(#) $Id: vstcoolServer.cdt,v 1.4 2003/07/14 12:38:56 vltscm Exp $"
//
// vstcoolServer.cdt
//
// who      when      what
// -----  -
// schipani 07/07/02  created
//-----
-

PUBLIC_COMMANDS

//=====
// general: Implemented by "too" module
//=====
COMMAND=  VERBOSE
SYNONYMS= lcuSetVerbose, SETVERB
FORMAT=  A
PARAMETERS=
    PAR_NAME=    onOff
    PAR_UNIT=
    PAR_TYPE=    STRING
    PAR_RANGE=   ENUM "On" , "Off"
    PAR_DEF_VAL= "Off"
REPLY_FORMAT= A
HELP_TEXT=
Switch on or off verbose mode. Verbose On gives multiple replies to commands
which take long time to execute. Verbose Off gives always only one reply
on completion.
@

COMMAND=  GETVERB
SYNONYMS= lcuGetVerbose
FORMAT=  A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get Verbose mode.
@

COMMAND=  DEBUG
SYNONYMS= lcuSetDebug, SETDEBU
FORMAT=  A
PARAMETERS=
    PAR_NAME=    onOff
    PAR_UNIT=
    PAR_TYPE=    STRING
    PAR_RANGE=   ENUM "On" , "Off"
    PAR_DEF_VAL= "Off"
REPLY_FORMAT= A
HELP_TEXT=
Switch on or off debugging mode. Debug On displays on LCU console development
```

debugging info.
@

COMMAND= GETDEBU
SYNONYMS= lcuGetDebug
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get Debugging mode.
@

COMMAND= TRACE
SYNONYMS= lcuSetTrace, SETTRA, SETTRAC
FORMAT= A
PARAMETERS=
 PAR_NAME= onOff
 PAR_UNIT=
 PAR_TYPE= STRING
 PAR_RANGE= ENUM "On" , "Off"
 PAR_DEF_VAL= "Off"
REPLY_FORMAT= A
HELP_TEXT=
Switch on or off trace mode. Trace On displays on LCU console development tracing info.
@

COMMAND= GETTRA
SYNONYMS= lcuGetTrace, GETTRAC
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get Tracing mode.
@

//=====

// LCC standard commands

//=====

COMMAND= INIT
SYNONYMS= vstcoolStdInit
FORMAT= A
PARAMETERS=
 PAR_NAME= devName
 PAR_UNIT=
 PAR_TYPE= STRING
 PAR_RANGE=
 PAR_DEF_VAL= "vstcool"
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. INIT initializes the "vstcool" device.
This command will change the state into Lovstaded.
The parameter device has to be "vstcool" to be compatible with LCC Node Management commands.
The LCC command lccInitAll sends the command INIT to vstcoolServer.
@

COMMAND= STANDBY
SYNONYMS= vstcoolStdStandby
FORMAT= A

```
PARAMETERS=
  PAR_NAME=      devName
  PAR_UNIT=
  PAR_TYPE=      STRING
  PAR_RANGE=
  PAR_DEF_VAL=   "vstcool"
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. If the "vstcool" device is in state On-Line
this command will change into state Stand-By. The command is not allowed
if the device is in another state (Lovstaded).
The parameter device has to be "vstcool" to be compatible with LCC Node
Management
commands.
The LCC command lccStandby sends the command STANDBY to vstcoolServer.
@

COMMAND=  ONLINE
SYNONYMS= vstcoolStdOnline
FORMAT= A
PARAMETERS=
  PAR_NAME=      devName
  PAR_UNIT=
  PAR_TYPE=      STRING
  PAR_RANGE=
  PAR_DEF_VAL=   "vstcool"
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. If the "vstcool" device is in state Stand-By
this command will change into state On-Line. The command is not allowed
if the device is in another state (Loaded).
The parameter device has to be "vstcool" to be compatible with LCC Node
Management
commands.
The LCC command lccOnLine sends the command ONLINE to vstcoolServer.
@

COMMAND=  OFF
SYNONYMS= vstcoolStdOff
FORMAT= A
PARAMETERS=
  PAR_NAME=      devName
  PAR_UNIT=
  PAR_TYPE=      STRING
  PAR_RANGE=
  PAR_DEF_VAL=   "vstcool"
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. It always changes the device into state Lovstaded.
The parameter device has to be "vstcool" to be compatible with LCC Node
Management
commands.
The LCC command lccShutdown sends the commands STOP and OFF to vstcoolServer.
No actions on "vstcool".
@

COMMAND=  STOP
SYNONYMS= vstcoolStdStop
FORMAT= A
PARAMETERS=
  PAR_NAME=      devName
```

```
PAR_UNIT=
PAR_TYPE=    STRING
PAR_RANGE=
PAR_DEF_VAL= "vstcool"
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. Stops ongoing activities.
It does not change the device state.
The parameter device has to be "vstcool" to be compatible with LCC Node
Management
commands.
The LCC command lccShutdown sends the commands STOP and OFF to vstcoolServer.
No actions on "vstcool".
@

COMMAND=    EXIT
SYNONYMS=   vstcoolStdExit
FORMAT=     A
PARAMETERS=
    PAR_NAME=    devName
    PAR_UNIT=
    PAR_TYPE=    STRING
    PAR_RANGE=
    PAR_DEF_VAL= "vstcool"
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. Terminates the application.
It always changes the device into state Off.
The parameter device has to be "vstcool" to be compatible with LCC Node
Management
commands.
No actions on "vstcool".
@

COMMAND=    SIMULAT
SYNONYMS=   vstcoolStdStartSim, STASIM
FORMAT=     A
PARAMETERS=
    PAR_NAME=    devName
    PAR_UNIT=
    PAR_TYPE=    STRING
    PAR_RANGE=
    PAR_DEF_VAL= "vstcool"
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. It updates the simulation state.
The parameter device has to be "vstcool" to be compatible with LCC Node
Management
commands.
No actions on "vstcool".
@

COMMAND=    STOPSIM
SYNONYMS=   vstcoolStdStopSim, STOSIM
FORMAT=     A
PARAMETERS=
    PAR_NAME=    devName
    PAR_UNIT=
    PAR_TYPE=    STRING
    PAR_RANGE=
    PAR_DEF_VAL= "vstcool"
```

REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command. It updates the simulation state and changes the general state into Lovstaded.
The parameter device has to be "vstcool" to be compatible with LCC Node Management commands.
No actions on "vstcool".
@

COMMAND= SELFTEST
SYNONYMS= vstcoolStdSelfTest
FORMAT= A
PARAMETERS=
 PAR_NAME= devName
 PAR_UNIT=
 PAR_TYPE= STRING
 PAR_RANGE=
 PAR_DEF_VAL= "vstcool"

REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command.
The parameter device has to be "vstcool" to be compatible with LCC Node Management commands.
No actions on "vstcool".
@

COMMAND= BREAK
SYNONYMS=
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command.
Break the "vstcool" server. Stops current activities. Analog to the STOP command.
@

COMMAND= KILL
SYNONYMS=
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Standard LCC command.
Kill the "vstcool" server. Analog to the EXIT command.
@

```
//=====
// general ...
//=====
COMMAND= VERSION
SYNONYMS= vstcoolStdGetVersion, GETVER
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get version of "vstadrot" Software.
@
```

COMMAND= STATE
SYNONYMS= vstcoolStdGetState, GETSTAT
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get LCC standard "vstcool" state.

OFF: the software device is not operational.

LOADED: the software is lovstaded and initialized.
the hardware is revstady to be initialized.

STANDBY: the software is lovstaded and initialized.
the hardware is initialized but in standby
(applicable parts switched off, brakes
clamped etc.)

ONLINE: the software is lovstaded and initialized.
the hardware is initialized and
fully operational.

@

COMMAND= STATUS
SYNONYMS= vstcoolStdGetStatus
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get "vstcool" substate.

IDLE: vstcool is ...

MONITORING: vstcool is ...

ERROR: while ... a fatal error occurred
@

COMMAND= GETSIM
SYNONYMS= vstcoolStdGetSimulation
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get LCC standard "vstcool" simulation state.
@

COMMAND= GETINIT
SYNONYMS= vstcoolStdGetInit
FORMAT= A
PARAMETERS=
REPLY_FORMAT= A
HELP_TEXT=
Get LCC standard "vstcool" initialisation state.
@

//=====

// Specific

//=====



//=====

//MAINTENANCE_COMMANDS
//

//=====

//TEST_COMMANDS

6.2 Command Interpreter Table

```

//*****
// O.A.C. - VST project
// #
// # "@(#) $Id: vstcoolServer.cit,v 1.4 2003/07/14 12:38:56 vltscm Exp $"
//
// vstcoolServer.cit
//
// who      when      what
// -----
// schipani 07/07/02  created
//-----

//=====
// Common to all modules: Implemented by "too" module
//=====
VERBOSE,IlcuSetVerbose, FUNCTION
DEBUG,  IlcuSetDebug,   FUNCTION
TRACE,  IlcuSetTrace,   FUNCTION
GETVERB,IlcuGetVerbose, FUNCTION
GETDEBU,IlcuGetDebug,   FUNCTION
GETTRA, IlcuGetTrace,   FUNCTION

//=====
// Common to all modules: LCC standard commands
//=====
INIT,    IvstcoolStdInit,          TASK, vstcoolInitTask,,,140000,REGISTER
STANDBY,IvstcoolStdStandby,       TASK, vstcoolStandbyTask,,,140000,REGISTER
ONLINE, IvstcoolStdOnline,        TASK, vstcoolOnlineTask,,,140000,REGISTER
OFF,    IvstcoolStdOff,           TASK, vstcoolOffTask,,,140000,REGISTER
STOP,   IvstcoolStdStop,          TASK, vstcoolStopTask,,,140000,REGISTER
EXIT,   IvstcoolStdExit,          FUNCTION
SIMULAT,IvstcoolStdStartSim,      FUNCTION
STOPSIM,IvstcoolStdStopSim,       FUNCTION
SELFTST,IvstcoolStdSelfTest,      FUNCTION

//=====
// General
//=====
VERSION, IvstcoolStdGetVersion,    FUNCTION
STATE,   IvstcoolStdGetState,      FUNCTION
GETSIM,  IvstcoolStdGetSimulation, FUNCTION
GETINIT, IvstcoolStdGetInit,       FUNCTION
STATUS,  IvstcoolStdGetStatus,     FUNCTION

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

//=====
// Test
//=====

```

7 Database Configuration

Each LCU has its own specific *vstcoolXXX.db* file to configure the *vstcool* db branch. All the db branches are built upon the general *vstcool* class.

7.1 vstcool.class

```

//*****
/** E.S.O. - VLT project
/**
/** "@(#) $Id: vstcool.class,v 1.3.1.3 2004/07/29 08:34:00 vltscm Exp $"
/**
/** who      when      what
/** -----
/** schipani 17/06/03  created - class to collect information from ESO
/**                               cabinet cooling controller
/** schipani 18/06/03  alarms added
/** schipani 10/05/04  alarm FATAL -> WARNING
/**

```

```

#ifndef VSTCOOL_CLASS
#define VSTCOOL_CLASS

```

```

CLASS BASE_CLASS vstcool
BEGIN

```

```

    ATTRIBUTE int32 State      2

    ATTRIBUTE double CabinetTemp  25
    ATTRIBUTE double AmbientTemp  25
    ATTRIBUTE double T_Inlet
    ATTRIBUTE double T_Outlet
    ATTRIBUTE int16 Flow1
    ATTRIBUTE int16 Flow2
    ATTRIBUTE int16 Flow3
    ATTRIBUTE int32 Cabinet_high
    ATTRIBUTE int32 Outlet_high
    ATTRIBUTE int32 Inlet_high
    ATTRIBUTE int32 Airflow_insuff
    ATTRIBUTE int32 Coolant_insuff

```

```

#ifndef MAKE_VXWORKS

```

```

    almAnalog2st("CabinetTemp", CabinetTemp, \
                NORMAL, 26, \
                WARNING, \
                3, 1, \
                SCOPE_LOCAL, 0)

```

```

#endif

```

```

END

```

```

#endif

```

7.2 vstcoolAZ.db

```

/*****
/** E.S.O. - VLT project
/**
/** "@(#) $Id: vstcoolAZ.db,v 1.3.1.3 2004/07/29 08:34:07 vltscm Exp $"
/**
/** who      when      what
/** -----
/** schipani 07/07/02  created
/** schipani 17/06/03  .class created
/** schipani 27/02/04  doubled the branches to handle 2 controllers
/** schipani 31/03/04  /iser device names added
/** brescia  15/07/04  3nd branch added
/**

#include "vstcool.class"

POINT BASE_CLASS :ESOCOOL
BEGIN
    ATTRIBUTE vstcool DATA1
    ATTRIBUTE vstcool DATA2
    ATTRIBUTE vstcool DATA3

        ATTRIBUTE BYTES32 DATA1ch      "/iser2"
        ATTRIBUTE BYTES32 DATA2ch      "/iser3"
        ATTRIBUTE BYTES32 DATA3ch      "/iser4"
END

```

7.3 vstcoolALT.db

```

/*****
/** E.S.O. - VLT project
/**
/** "@(#) $Id: vstcoolALT.db,v 1.3.1.3 2004/07/29 08:34:05 vltscm Exp $"
/**
/** who      when      what
/** -----
/** schipani 07/07/02  created
/** schipani 17/06/03  .class created
/** schipani 27/02/04  doubled the branches to handle 2 controllers
/** schipani 31/03/04  /iser device names added
/** brescia  15/07/04  3nd branch added
/**

#include "vstcool.class"

POINT BASE_CLASS :ESOCOOL
BEGIN
  ATTRIBUTE vstcool DATA1
  ATTRIBUTE vstcool DATA2
  ATTRIBUTE vstcool DATA3

  ATTRIBUTE BYTES32 DATA1ch    "/iser2"
  ATTRIBUTE BYTES32 DATA2ch    "/iser3"
  ATTRIBUTE BYTES32 DATA3ch    "/iser4"
END

```

7.4 vstcoolADC.db

```
/** *****  
/** E.S.O. - VLT project  
/**  
/** "@(#) $Id: vstcoolADC.db,v 1.3.1.3 2004/07/29 08:34:03 vltscm Exp $"  
/**  
/** who      when      what  
/** -----  
/** schipani 07/07/02  created  
/** schipani 17/06/03  .class created  
/** schipani 27/02/04  doubled the branches to handle 2 controllers  
/** schipani 31/03/04  /iser device names added  
/** brescia  15/07/04  3nd branch added  
/**  
  
#include "vstcool.class"  
  
POINT BASE_CLASS :ESOCOOL  
BEGIN  
    ATTRIBUTE vstcool DATA1  
    ATTRIBUTE vstcool DATA2  
  
    ATTRIBUTE BYTES32 DATA1ch    "/iser2"  
    ATTRIBUTE BYTES32 DATA2ch    "/iser3"  
    ATTRIBUTE BYTES32 DATA3ch    ""  
END
```

7.5 vstcoolM12.db

```
/** *****  
/** E.S.O. - VLT project  
/**  
/** "@(#) $Id: vstcoolM12.db,v 1.3.1.3 2004/07/29 08:34:09 vltscm Exp $"  
/**  
/** who      when      what  
/** -----  
/** schipani 07/07/02  created  
/** schipani 17/06/03  .class created  
/** schipani 27/02/04  doubled the branches to handle 2 controllers  
/** schipani 31/03/04  /iser device names added  
/** brescia  15/07/04  3nd branch added  
/**  
  
#include "vstcool.class"  
  
POINT BASE_CLASS :ESOCOOL  
BEGIN  
    ATTRIBUTE vstcool DATA1  
    ATTRIBUTE vstcool DATA2  
  
        ATTRIBUTE BYTES32 DATA1ch      "/iser2"  
        ATTRIBUTE BYTES32 DATA2ch      "/iser3"  
        ATTRIBUTE BYTES32 DATA3ch      ""  
END
```

8 Configuration files

The scan links are not managed by dedicated files locally in the module in order to lessen the total number of scan link files required in the whole TCS package (because of a limitation of the `pkgin` utility, which currently limits to 30 the maximum number of scan files). The scanned data related to `vstcool` module are therefore configured in the installation module `vsttcsINS lcu .links` files. As an example the configuration for azimuth (extracted from `vsttcsINS az .links` file) is:

```
# ESO cabinet cooling controller 1 data
:Appl_data:TCS:LCU:az:ESOCOOL1.State          :ESOCOOL:DATA1.State          SRBX 2  dbINT32      1

:Appl_data:TCS:LCU:az:ESOCOOL1.CabinetTemp   :ESOCOOL:DATA1.CabinetTemp   SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL1.AmbientTemp   :ESOCOOL:DATA1.AmbientTemp   SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL1.T_Inlet       :ESOCOOL:DATA1.T_Inlet       SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL1.T_Outlet      :ESOCOOL:DATA1.T_Outlet      SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL1.Flow1        :ESOCOOL:DATA1.Flow1        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL1.Flow2        :ESOCOOL:DATA1.Flow2        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL1.Flow3        :ESOCOOL:DATA1.Flow3        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL1.Cabinet_high  :ESOCOOL:DATA1.Cabinet_high  SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL1.Outlet_high   :ESOCOOL:DATA1.Outlet_high   SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL1.Inlet_high    :ESOCOOL:DATA1.Inlet_high    SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL1.Airflow_insuff :ESOCOOL:DATA1.Airflow_insuff SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL1.Coolant_insuff :ESOCOOL:DATA1.Coolant_insuff SRBX 2  dbINT32      1

# ESO cabinet cooling controller 2 data
:Appl_data:TCS:LCU:az:ESOCOOL2.CabinetTemp   :ESOCOOL:DATA2.CabinetTemp   SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL2.AmbientTemp   :ESOCOOL:DATA2.AmbientTemp   SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL2.T_Inlet       :ESOCOOL:DATA2.T_Inlet       SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL2.T_Outlet      :ESOCOOL:DATA2.T_Outlet      SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL2.Flow1        :ESOCOOL:DATA2.Flow1        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL2.Flow2        :ESOCOOL:DATA2.Flow2        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL2.Flow3        :ESOCOOL:DATA2.Flow3        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL2.Cabinet_high  :ESOCOOL:DATA2.Cabinet_high  SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL2.Outlet_high   :ESOCOOL:DATA2.Outlet_high   SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL2.Inlet_high    :ESOCOOL:DATA2.Inlet_high    SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL2.Airflow_insuff :ESOCOOL:DATA2.Airflow_insuff SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL2.Coolant_insuff :ESOCOOL:DATA2.Coolant_insuff SRBX 2  dbINT32      1

# ESO cabinet cooling controller 3 data
:Appl_data:TCS:LCU:az:ESOCOOL3.CabinetTemp   :ESOCOOL:DATA3.CabinetTemp   SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL3.AmbientTemp   :ESOCOOL:DATA3.AmbientTemp   SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL3.T_Inlet       :ESOCOOL:DATA3.T_Inlet       SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL3.T_Outlet      :ESOCOOL:DATA3.T_Outlet      SRBX 2  dbDOUBLE      1
:Appl_data:TCS:LCU:az:ESOCOOL3.Flow1        :ESOCOOL:DATA3.Flow1        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL3.Flow2        :ESOCOOL:DATA3.Flow2        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL3.Flow3        :ESOCOOL:DATA3.Flow3        SRBX 2  dbUINT16     1
:Appl_data:TCS:LCU:az:ESOCOOL3.Cabinet_high  :ESOCOOL:DATA3.Cabinet_high  SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL3.Outlet_high   :ESOCOOL:DATA3.Outlet_high   SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL3.Inlet_high    :ESOCOOL:DATA3.Inlet_high    SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL3.Airflow_insuff :ESOCOOL:DATA3.Airflow_insuff SRBX 2  dbINT32      1
:Appl_data:TCS:LCU:az:ESOCOOL3.Coolant_insuff :ESOCOOL:DATA3.Coolant_insuff SRBX 2  dbINT32      1
```

9 Abbreviation and Acronyms

ADC	Adapter/Cassegrain
AD/ROT	Adapter/Rotator
AG	Auto-Guider
ALT	Altitude
AO	Active Optics
AZ	Azimuth
CCS	Central Common Software
CDT	Command Definition Table
CI	Command Interpreter
CIT	Command Interface Table
DB	Database
ESO	European Southern Observatory
GUI	Graphical User Interface
IA	Image Analysis
LCC	LCU Common Software
LSF	LCU Server Framework
LCU	Local Control Unit
OAC	Capodimonte Astronomical Observatory
TCS	Telescope Control Software
TIF	Telescope Interface
TWG	Technology Working Group
VLT	Very Large Telescope
VST	Very Large Telescope Survey Telescope
WS	Workstation

10 Pendings

The WHOLE document describes a software not officially released and in modification phase. Therefore many modifications are going to be done on this document in the next months till the completion of the project.

__oOo__