EUROPEAN SOUTHERN OBSERVATORY

Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

INSTRUMENTATION DIVISION

New General detector Controller

Optical DCS - User Manual

Document Number: VLT-MAN-ESO-13660-4086

Document Issue: 9.0

Date of Issue: 15/05/2015

Author :	Name	Date	Signature
	Claudio Cumani	15/5/2015	
Job Manager :	Name	Date	Signature
	Bruno Gilli	19/5/2015	
Releaser :	Name	Date	Signature
	Jochen Haucke	20/5/15	

CHANGE RECORD

ISSUE	DATE	SECTIONS AFFECTED	REASON/INITIATION DOCUMENTS/REMARKS
0.1	22-09-2006	All	First draft, basic information for optical prototype
1.0	31-07-2007	All	Document widely rewritten, on the base of the comments to the first draft and of the NGCOSW implementation
2.0	31-08-2007	1.1 2.1 3.1.1 3.1.1 3.2.3.2 4 6.1 6.1 7 9.1.1 9.1.2 9.1.2 9.1.4 9.1.5	Optical exposure loop description corrected ngcoctr module added Local-Hardware-Test renamed Local-Software-Test Operational mode macros are defined in ngco.h Configuration values updated Command interface table updated List of public online database attributes updated NOTE added SETUP can be issued also for paused exposure. Exposure types macros are defined in ngco.h Description of exposure status values improved Exposure status macros are defined in ngco.h NGCOSW creates exposure Id, if not passed at START Description of differences with FIERA corrected
3.0	18-06-2008	1 2 2.1 2.4 2.5 3.1.1 3.2.1 3.2.1 3.2.1 3.2.3.2 3.2.3.2 3.2.3.2 4 4.1 7.1 7.1 (prev. 7.2) 13 14 15.2 16	Reference to DOORS obsolete Removed NGC base package modules Updated list of NGCOSW modules Online database environment generation added INS_ROOT population added Configuration Set updated “Software-Test” mode removed DET.CON.GUI follows IR definition DET.CON.XTERM added -xterm option added -kill option removed “Local-Software-Test mode” removed Command interface table updated Changes with respect to FIERA updated “Complete Setup” chapter removed (obsolete) List of changes updated rtd interface description updated Description of GUIs added Evaluation of T/P implementation added STANDBY and exposure status readout added

4.0	30-10-2008	2.3 (prev 2.4)	Oldb generation split btw IWS and LLCU cases
		2.3.1	Online database environment generation added
		2.3.2	Online database environment generation added
		2.3.3	Online database verification added
		2.3.4	Online database startup added
		2.3.5	Detector user description added
		2.4	Logging configuration added
		2.5	INS_ROOT population description improved
		3.2.3.2	Example updated
		16	Manpage Chapter added
		17 (prev. 16)	Minor corrections
5.0	26-02-2010	1.10	Acknowledgments added
		2.1	NGCOSW module list updated
		3.2.1	Startup option for image post-processing added
		3.2.3.1	<xx>dcfgCONFIG.cfg updated
		3.2.3.2	<xx>dcfgDCS.cfg added
		3.2.3.3	<xx>dcfgCAMERA.cfg updated
		5.1	Customization of Super Control process added
		6.3	Image post-processing interface description added
		9.6	Image post-processing description added
		14.1	Fig. 5 (Optical NGC Control Panel) updated
		14.2	Fig. 6 (NGC Engineer Panel) updated
		16	Manpages updated
		16.1.7	ngcoDcsTemplate manpage added
6.0	05-05-2010	2.2.2	Info for pkgin configuration added
		2.3.1	Description improved
		2.5.1	Multi-DCS instrument case adedd
		3.2.1	Startup Configuration chapter added
		3.2.3.4	Super Control example configuration file added
		5	NOTE for the instrument configuration file added
		5.2	Startup of customized Super Control process added
		5.2	Usage of DET.FRAM.MERGE added
		16.1.2	ngcoDcsInstall manpage updated
		16.1.3	ngcoDcsStart manpage updated
7.0	01-10-2010	3.2.3.1	<xx>dcfgCONFIG.cfg updated
		3.2.3.2	<xx>dcfgDCS.cfg updated
		3.2.3.3	<xx>dcfgCAMERA.cfg updated
		3.5	New paragraph on editable configuration keywords
		5.1	Description improved
		5.4	Reference to shutter synchronization added
		6.1	Oldb attribute list updated
		9.1.1	Exposure type descriptions corrected
		9.2	Warning on WAIT command behavior added
		10	Chapter divided in paragraphs
		10.4	Shutter synchronization added
		App. A	LLCU IP and hostname reconfiguration notes added

8.0	17-07-2012	All	SVN replaces CMM
		All	NGCOPT replaces NGCOSW
		1.1	“merging” exposure state added
		2.0	Base VLTSW version updated
		2.1	ngcoarc removed
		2.2.1	Removed: only pkgin installation used
		2.2.2	Removed: only pkgin installation used
		2.3.1	DATABASE.db examples have been updated
		2.3.4	Automatic environment startup modified
		2.5	Scan system configuration added
		3.2.3.2	<xx>dcfgDCS.cfg updated
		3.2.3.3	<xx>dcfgCAMERA.cfg updated
		3.2.3.4	Super Control Process Configuration Set updated
		4	STARTWP/STOPWP implemented
		4	STARTTL/STOPTL removed
		5.3	Example added
		6.1	Oldb attribute list updated
		7	Keyword names updated (SPR VLTSW20100252)
		7 – Table 10	DET.FRAM.FITSMTD implemented
		7 – Table 10	DET.READ.NFRAM implemented
		7 – Table 10	Single windowing implemented
		7	Usage of DET.MODEi.WINSUP explained
		7.1	DET.FRAM.FITSMTD is NOT obsolete
		8	STATUS is a command for all NGCOSW processes
		9	Setup keywords updated (SPR VLTSW20100252)
		9.1.2	MERGING exposure state added
		9.1.3	fileName attribute path corrected
		9.1.5	“Readout Types” paragraph added
		9.1.6	“Shutter Types and modes” paragraph added
		9.1.7	“Time Stamps” paragraph added
		9.1.8	MERGING exposure state added
		9.3	DET.READ.NFRAM implemented
		9.3	Data-cube format description corrected
		9.3	Merging for multi-DCS systems implemented
		9.5	Reference to ESO-VLT-DIC.PRIMARY-FITS removed
		9.5	NOTE removed
		9.5.1	Usage of .det file not required
		10.4	Shutter synchronization description improved
		11	Error definitions updated
		13	Usage of rtdcore added
		14.1 – Fig. 5	Updated
		15.1	Pulpo is NOT the default shutter controller
		15.2	NGCOPT does not support telemetry
		15.3	NGCOPT does not support Adaptive Optics
		16	All manpages updated
		App. A	Definitions of users performing different actions added
		App. B	Troubleshooting added

9.0	15-05-2015	2.3.5	Description of setting of .rhosts file added
		3.2.1 (Table 1)	Description of DET.CON.MRGPROC added
		3.2.5	DETi.MODEm.PREAMPn.BNDWTHi added
		3.2.3.2	Example of DCS.cfg file updated
		3.2.3.3	Example of CAMERA.cfg file updated
		3.2.3.4	Example of Super Control cfg file updated
		5.2	Usage of DET.CON.MRGPROC added
		6.1 (Table 4)	system.opmode attribute corrected in system.opMode
		6.1 (Table 4)	exposure:data.binx / biny attributes added
		7 (Table 6)	DET.SHUT.ENABLE, DET.PFLASHi.EXPTIME added
		7 (Table 7)	DET.EXP.TIMEREP, DET.EXP.WIPETIME implemented
		7 (Table 10)	DET.FRAM.OFFSET, DET.RTD1.REFRESH added
		7 (new Tab. 11)	DET.CON.LOG, DET.OPT.VRELOAD, DET.WIPE.TIMEREP added
		9.1.1	Led and LedShut exposure types implemented
		9.1.5	SlowMosaicOffset, SlowMosaicOffsetSimple types added
		(new) 9.1.7	Description of usage of external LEDs added, difference between Preflash and FlatField LEDs explained.
		11 (Table 13)	Error list updated
		14.1 (Fig. 5)	Optical NGC Control Panel updated
		14.1. (Fig. 6)	Subpanel for Preflash time setting added
		14.1 (Fig. 7)	NGC Control Panel for Super Control Process added
		Appendix B	Startup error cases added

Table of contents

1. Introduction 10
1.1. Purpose 10
1.2. Scope 11
1.3. Applicable Documents 11
1.4. Reference Documents 11
1.5. Abbreviations and Acronyms 11
1.6. Glossary 11
1.7. Stylistic Conventions 11
1.8. Naming Conventions 12
1.9. Problem Reporting/Change Request 12
1.10. Acknowledgments 12
2. Installation and configuration 13
2.1. Software Modules 13
2.2. Installation 13
2.3. Online database environment 14
2.3.1. IWS online database generation 14
2.3.2. NGC-LLCU online database generation 15
2.3.3. Online database verification 16
2.3.4. Online database automatic startup 16
2.3.5. Detector user 17
2.4. NGC-LLCU logging system configuration 17
2.5. IWS scan system configuration 17
2.6. INS_ROOT population 18
2.6.1. INS_ROOT population for multi-DCS instruments 18
3. Startup/Shutdown Procedure 19
3.1. System Configuration 19
3.1.1. NGCOPT operational modes 21
3.1.2. NGC General Purpose Control Server operational modes 21
3.1.3. Simulation of the NGC detector electronics 22
3.2. System Startup 22
3.2.1. Startup Configuration 22
3.2.2. Startup Procedure 24
3.2.3. Changes with respect to FIERA 25
3.2.4. Configuration Examples 26
3.2.3.1 Startup Configuration file <label>CONFIG.cfg 26
3.2.3.2 Configuration Set <label>DCS.cfg 27
3.2.3.3 Configuration Set <label>CAMERA.cfg 29
3.2.3.4 Super Control Process Configuration Set 34
3.2.5. Editable keywords in configuration files 36

3.3. NGCOPT operational states 37
3.3.1. Changes with respect to FIERA 38
3.4. System Shutdown 38
3.4.1. Changes with respect to FIERA 38
4. Command Interface 39
4.1. Changes with respect to FIERA 41
5. Multiple Instances of DCS 42
5.1. Customization of the Super Control process 42
5.2. Startup of a system using multiple instances of DCS 43
5.2.1. How to run the standard Super Control process 43
5.2.2. How to run the customized Super Control process 43
5.3. How to modify the number of instances of DCS 43
5.4. Shutter synchronization 45
6. Database Interface 46
6.1. Interface between NGCOPT and the external environment 46
6.2. Interface between NGCOPT and TCS 48
6.3. Image post-processing interface 48
6.4. ngcoDbPublic.h 49
6.5. Changes with respect to FIERA 49
7. Setup Command 50
7.1. Changes with respect to FIERA 54
8. Status Command 55
8.1. Changes with respect to FIERA 55
9. Exposure Handling 56
9.1. Description 56
9.1.1. Exposure types 56
9.1.2. Exposure status 56
9.1.3. Image data 58
9.1.4. Exposure Id 58
9.1.5. Readout types 58
9.1.6. Shutter types and modes 58
9.1.7. External LEDs 59
9.1.8. Time stamps 60
9.1.9. Changes with respect to FIERA 60
9.2. Commands 60
9.2.1. Changes with respect to FIERA 61
9.3. File Formats 61
9.3.1. Changes with respect to FIERA 62
9.4. Naming Schemes 62
9.4.1. Changes with respect to FIERA 62

9.5. FITS-Header Contents 62
9.5.1. Changes with respect to FIERA 62
9.6. Image post-processing 62
9.6.1. Creating a module which implements post-processing 63
9.6.2. How to run the process which implements post-processing 63
10. Synchronization 66
10.1. Synchronization via external trigger 66
10.2. Synchronization via global run-signal 66
10.3. Synchronization via system time 66
10.4. Synchronization with shutter operations 66
11. Error Definitions 68
12. Error and Logging Handling 73
13. Real-Time Display Interface 74
13.1. Changes with respect to FIERA 74
14. Graphical User Interface 75
14.1. Control Panel 75
14.2. Engineering Interface 79
15. Special functionalities for Optical Instruments 80
15.1. Shutter Control by Pulpo Server 80
15.2. Temperature/pressure Monitoring 80
15.3. Adaptive Optics 80
16. Manpages 81
16.1.1. ngcoDcsOldb 81
16.1.2. ngcoDcsInstall 83
16.1.3. ngcoDcsStart 86
16.1.4. ngcoDcsStop 89
16.1.5. ngcoGetProcNum 91
16.1.6. ngcoDcsClean 92
16.1.7. ngcoDcsTemplate 93
17. Example of NGCOPT usage 94
Appendix A. Change LLCU’s address and hostname 96
Appendix B. Troubleshooting 98

List of Figures

Figure 1 - Optical NGC software Architecture 20
Figure 2 - Operational states and state transitions 38
Figure 3 - Post Processing: Image Transfer Client class 64
Figure 4 - Post Processing: Image Transfer Client sequence diagram 65
Figure 5 - Optical NGC Control Panel 76
Figure 6 - Subpanel for Preflash time setting 77
Figure 7 - Optical NGC Control Panel for the Super Control Process 78
Figure 8 - NGC Engineer Panel 79

List of Tables

Table 1 - Startup Configuration Keywords 24
Table 2 - Command list 40
Table 3 - Special command list 41
Table 4 - Online database attributes for detector system monitoring 48
Table 5 - Online database attributes for TCS 48
Table 6 - Basic Setup keywords for single exposure 51
Table 7 - Additional Setup keywords for loops of exposures 52
Table 8 - Setup keywords for multistep exposures 52
Table 9 - Setup keywords for windowing 52
Table 10 - Setup keywords for image display 53
Table 11 - Setup keywords for miscellaneous purposes 54
Table 12 - Errors common to all NGCOPT modules 69
Table 13 - Errors specific to the different NGCOPT modules 71

1. Introduction

The software described in this manual is intended to be used in the ESO VLT project by ESO and authorized external contractors only.

While every precaution has been taken in the development of the software and in the preparation of this documentation, ESO assumes no responsibility for errors or omissions, or for damage resulting from the use of the software or of the information contained herein.

1.1. Purpose

This document is the User Manual of the New Generation detector Controller (NGC) Control Software for optical instruments (NGCOPT).

It is intended to provide people, who intend to use the NGC Controller for optical Instruments, with all the necessary information to **install** from scratch the NGCOPT, **interact programmatically** with the NGCOPT, operate an optical camera as a simple **standalone** instrument.

The manual assumes that the reader has some knowledge of C/C++ and Tcl/Tk languages, UNIX Operating System, VLT Software, in particular CCS. It is not intended to be an introduction to optical CCD cameras, and therefore it uses common terminology in this field (e.g. pixel, binning, readout, frame-transfer chip, etc.) without further explanation.

The control software for infrared applications (NGCIRSW) is described in a separate manual [RD77]. Basically the NGC electronics [AD8] is the same for both infrared and optical applications. Nevertheless there are many differences concerning the usage of the controller and the data acquisition and data handling procedures. To cover both applications in an effective way and also to have a certain backwards compatibility with the predecessors FIERA and IRACE, different SW-architectures have been chosen, which are described in detail in the NGC SW design documents [AD9], [AD10] and [AD11]. The following paragraph summarizes the main differences:

- **Detector Read-Out Schemes**

  For an infrared detector (CMOS / non destructive readout) the clock-pattern generation runs in an infinite loop and the detector is read-out/reset all the times. An optical detector (CCD / destructive readout) is read-out just once at the end of an exposure.

- **Data Handling**

  The optical application delivers one frame at the end of the exposure and the only processing to be done is pixel sorting, centroiding and possibly and offset correction (if not yet done in HW). The infrared data require some pre-processing depending on the read-out mode of the detector in use. The read-out modes, the pre-processing algorithms and the setup-parameters for these algorithms are manifold and require a very high degree of flexibility. The pre-processing task produces an arbitrary number of different result frame types, which all have to be transferred and/or displayed on demand. This also has an impact on the RTD-interface.

- **Exposure Loops**

  For infrared applications *starting an exposure* basically means starting to transfer the acquired data to a FITS-file (i.e. the server has to attach to and keep step with a running procedure). The end-of-exposure condition is flexible and depends on both the requested frame types and on the number of frames of each type to be produced and stored. The optical exposure always terminates with the saving of the data, which are read at the end of the exposure and follows a much more rigid scheme (“*inactive*” - “*wiping*” [- “*pending*”] - *integrating*” - “*reading*” - “*transferring*” [- *“merging’*] - “*inactive*”). This scheme implies an active intervention of the control-server during the exposure like the application of new voltages in each state and the additional shutter-control, whereas the infrared control-server mainly reacts passively on incoming data-frames once the exposure is started. So basically the demands on process concurrency are very different in both cases.

A conscious effort has been made to maintain a certain degree of backwards compatibility of NGCOPT with FIERASW. Where applicable, a hint to the major changes with respect to FIERASW can be found at the end of each section.

1.2. Scope

Scope of this document is the NGC Control Software for optical instruments (NGCOPT).

1.3. Applicable Documents

Applicable documents used in the NGC project are listed in the document VLT-LIS-ESO-13660-3906 "NGC Project Documentation".

1.4. Reference Documents

Reference documents used in the NGC project are listed in the document VLT-LIS-ESO-13660-3906 "NGC Project Documentation".

1.5. Abbreviations and Acronyms

Abbreviations and acronyms used in the NGC project are listed in [RD64].

1.6. Glossary

All the relevant concepts used within the NGC project are listed in [RD63].

1.7. Stylistic Conventions

The following styles are used:

bold in the text, for commands, filenames, pre/suffixes as they have to be typed.

italic in the text, for parts that have to be substituted with the real content before typing.

courier for examples, commands, filenames as they have to be typed.

<name> in the examples, for parts that have to be substituted with the real content

The **bold** and *italic* styles are also used to highlight words.

1.8. Naming Conventions

This implementation follows the naming conventions as outlined in [AD27].

1.9. Problem Reporting/Change Request

The form described in [AD72] shall be used.

1.10. Acknowledgments

Special thanks to Andrea Balestra for his fundamental contribution to the development of the NGCOPT and Luigi Andolfato for his precious support and suggestions on `wsf`.

2. Installation and configuration

NGCOPT runs on ESO standard Instrument Workstations (IWS) and NGC-LLCUs[1].

The VLTSW (version 2010 or more recent) must have been already installed on the hosts.

2.1. Software Modules

The NGC optical detector control software package (***NGCOPT***) package consists of:

- ***ngco*** - This module contains all required scripts for system startup and shutdown, plus a set of utilities.
- ***ngcocon*** - The NGC system coordination module for optical applications. This includes all required scripts for system startup and shutdown.
- ***ngcoctr*** - The NGC exposure Control module for optical applications.
- ***ngcoexp*** - The NGC Exposure Coordination module for optical applications.
- ***ngcoits*** - The NGC Image Transfer Server module for optical applications.
- ***ngcoitc*** - The NGC Image Transfer Client module for optical applications.
- ***ngcosc*** - The NGC Super Control module for optical applications. This process is needed only for instruments controlling multiple DCS instances (see 5).
- ***ngcoui*** - Engineering GUI used for direct system interaction and data acquisition.

All software modules are under SVN configuration control.

2.2. Installation

Retrieve from the archive the appropriate NGCOPT tag:

```
svn co $SVNREPO/tags/RC/NGC/NGCOPT-<tag_id>
```

The `ngcins` software module contains a pkgin installation-configuration (**for both NGC IR and OPT software**):

```
cd NGCOPT-<tag_id>
pkginBuild ngcins
```

Module ngcins contains a number of modules and keyword OPTIONS in the pkgin configuration has to be set to “`LIBRARY`” as shown in the following example:

```
INSTALL.MODULE5.NAME      "ngcins";
INSTALL.MODULE5.SUBPKG    "NGC";
```

[1] ESO Standard hardware is described in http://websqa.hq.eso.org/sdd/bin/view/SDDInfo/LinuxStandardHw

```
INSTALL.MODULE5.OPTIONS   "LIBRARY";
INSTALL.MODULE9.MODULES   "dicNGC ngcb ngcdcs ngcdrv ngcgui
ngciracq ngcircon ngcpp ngcrtd ngco ngcocon ngcoctr ngcoexp
ngcoitc ngcoits ngcosc ngcoui";
```

2.3. Online database environment

To automatically generate the online database, the environment variables RTAPENV, CCDLENV and CCDNAME must be defined:

- RTAPENV defines the name of the local online database environment
- CCDLENV on the IWS defines the name of the remote online database environment, on the NGC-LLCU it must be set to 0
- CCDNAME defines the name of the detector camera

NOTE : on the NGC-LLCU the environment variables are defined in the files

```
/etc/pecs/releases/000/etc/locality/apps-all.env
/etc/pecs/releases/000/etc/locality/apps-${HOST}.env
```

On the IWS you could define them in the same files or in

```
~/.pecs/apps-${HOST}.env
```

The script ngcoDcsOldb performs a preliminary system check: if the environment variables are defined, if the ACC server is defined and running, if local and remote environments are defined on the local computer and in the ACC server, if the scanning has been properly configured, if the user which shall run the software is defined on the local and the remote computer, etc. (manpage of ngcoDcsOldb is available in 16.1.1.)

The same script then handles the online database generation, acting in different ways on the IWS (see 2.3.1) and the LLCU (see 2.3.2).

2.3.1. IWS online database generation

On the IWS, the DATABASE.db.NGCOPT and USER.db.NGCOPT templates for the online database environment are installed in the $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl directory by running:

```
ngcoDcsOldb -renv $CCDLENV -host IWS
```

(manpage of ngcoDcsOldb is available in 16.1.1.)

Use the DATABASE.db.NGCOPT and USER.db.NGCOPT templates to edit the DATABASE.db and USER.db files.

In the template DATABASE.db.NGCOPT it is described how to describe different configurations (instrument controlling only one camera, instrument controlling more cameras).

Examples:

- Instrument controlling one camera:

  Add in the DATABASE.db the following, replacing <myCCDNAME> with the name of the camera (e.g., $CCDNAME), <:myPATH> with the preferred oldb location (e.g., :DCS:optical):

  ```
  #define CCDNAME <myCCDNAME>
  #define NGCROOT :Appl_data<:myPATH>:CCDNAME
  #include “ngco.db”
  ```

- Instrument controlling four cameras:

  Add in the DATABASE.db the following, replacing:

  - <myINSTRUMENT> with the instrument name (i.e., the name of the branch for the Super Control process, see 5);
  - <myCCDNAME1>, <myCCDNAME2>, <myCCDNAME3> , <myCCDNAME4> with the camera names (i.e., the values of $CCDNAME on the different LLCUs), <:myPATH> with the preferred oldb location (e.g., :DCS:optical):

  ```
  #define DCSNAME <myINSTRUMENT>
  #define DCSROOT  :Appl_data<:myPATH>:DCSNAME

  #define CCDNAME1 <myCCDNAME1>
  #define NGCROOT1 :Appl_data<:myPATH>:DCSNAME:CCDNAME1

  #define CCDNAME2 <myCCDNAME2>
  #define NGCROOT2 :Appl_data<:myPATH>:DCSNAME:CCDNAME2

  #define CCDNAME3 <myCCDNAME3>
  #define NGCROOT3 :Appl_data<:myPATH>:DCSNAME:CCDNAME3

  #define CCDNAME4 <myCCDNAME4>
  #define NGCROOT4 :Appl_data<:myPATH>:DCSNAME:CCDNAME4

  #include “ngco.db”
  ```

Once the DATABASE.db and USER.db files have been properly edited, generate the environment: in $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl run

```
make clean db
```

To initialize and start the environment run:

```
vccEnvInit -e $RTAPENV

vccEnvStart -e $RTAPENV
```

2.3.2. NGC-LLCU online database generation

On the NGC-LLCU, the RTAPENV online database environment is automatically generated and started by running:

```
    ngcoDcsOldb -renv <IWS_RTAPENV> -host LLCU
```

Manpage of ngcoDcsOldb is available in 16.1.1.

2.3.3. Online database verification

On both the IWS and the LLCU verify that the environment has been generated:

```
    dbRead "<alias><myCCDNAME>:exposure:control.state"
```

replacing <myCCDNAME> with the name of the camera (e.g., $CCDNAME).

Verify that the needed processes are running in the CCS environment, by using `ccsPerfMon`. A view similar to the following should be displayed:

```
Process Name          |PNUM |  PID  |  UID  |  GID  |  MSGID   | MONPID
°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°
ccsScheduler          | 1   | 4224  | 3227  | 300   | -1       |    -1
ccsSHManager          | 2   | 4228  | 3227  | 300   | 1582563329 |    -1
qsemu                 | 3   | 4231  | 3227  | 300   | 1583415311 |    -1
evtEventConfg         | 4   | 4239  | 3227  | 300   | 1583087623 |    -1
timsTimeKeeper        | 5   | 4233  | 3227  | 300   | 1582825475 |    -1
scanMngr              | 6   | 4248  | 300   | 300   | 1583284234 |    -1
ccsScan               | 7   | 4259  | 300   | 300   | 1583480856 |    -1
ccsCmdServer          | 8   | 4243  | 3227  | 300   | 1583153160 |    -1
alrmServer            | 9   | 4247  | 3227  | 300   | 1583218697 |    -1
cmdManager            | 12  | 4236  | 3227  | 300   | 1582891012 |    -1
msgServer             | 13  | 4237  | 3227  | 300   | 1582956549 |    -1
logManager            | 14  | 4238  | 3227  | 300   | 1583022086 |    -1
alrmLogger            | 16  | 4268  | 3227  | 300   | 1583677452 |    -1
hisDHMngr             | 17  | 4269  | 300   | 300   | 1583743004 |  4270
ccsPerfMon            | 22  | 6532  | 3227  | 300   | 1589313576 |    -1
dbMQDBM               | 58  | 4232  | 3227  | 300   | 1582759938 |    -1
```

2.3.4. Online database automatic startup

On both the IWS and the LLCU the online database environment start automatically at boot by defining RTAPENV and RTAPRUNNER in the file `/etc/pecs/releases/000/etc/locality/apps-`hostname`.env`

e.g.:

```
        export RTAPENV=w`hostname`
        export RTAPRUNNER=<myUser>
```

where <myUser> is the user managing the online database.

NOTE: `<myUser>` defined as `RTAPRUNNER` must be the owner of the `/tmp/${RTAPENV}` file, the `$VLTDATA/ENVIRONMENTS/$RTAPENV` directory, subdirectories and files.

2.3.5. Detector user

In order for the online database to function correctly, the user which runs NGCOPT must be defined on both the IWS and the LLCU, with the same user id.

On the LLCU, the `$HOME/.rhosts` must contain the IWS hostname and the name of the detector user. The reason is that the `ngcoDcsStart` and `ngcoDcsStop` scripts perform remote shell commands from the IWS to the LLCU (see 16.1.3 and 16.1.4).

If (for development or testing) the complete NGCOPT SW must run on the IWS (i.e., with `$CCDLENV=$RTAPENV`, of course in Hardware-Simulation mode, see 3.1.1), the `$HOME/.rhosts` must contain the IWS hostname and the name of the detector user. The reason is that the `ngcoDcsStart` and `ngcoDcsStop` scripts perform remote shell commands, in this case from the IWS to the IWS itself (see 16.1.3 and 16.1.4).

2.4. NGC-LLCU logging system configuration

To configure the LLCU to log messages onto the IWS, edit the file `/etc/syslog.conf` while logged in as the user “root”.

```
# ===========================================================
# The following three lines configure the VLT logging system
# ===========================================================
#*info;mail,local1,local2.none /var/adm/messages
#local1.warning /vltdata/tmp/logFile
#local2.warning /vltdata/tmp/logAuto
*.info;mail,local1,local2.none @myIws
local1.warning @myIws
local2.warning @myIws
```

Substitute the IWS hostname for “`myIws`”.

IMPORTANT: use tabs for spacing!

This change will take effect after rebooting the LLCU, or run

```
     kill -HUP `ps -C syslogd -o pid=`
```

to restart the loggin daemon.

2.5. IWS scan system configuration

If more than one LLCU is used, modify the default amount of scan shared memory area to be allocated by editing the file

```
     /vltdata/ENVIRONMENTS/$RTAPENV/CcsEnvTable
```

and setting

```
          6   3   Y   N   N   Y   Y   30000   10    scanMngr -m <size>
```

where <size>=10000*number of LLCUs

2.6. INS_ROOT population

To automatically populate the INS_ROOT (instrument directory), the environment variable INS_ROOT must be defined, the directory `$INS_ROOT` must exist and the “instrument module” must have been installed (“instrument module” is the SVN module containing the detector startup configuration file <xx>`dcfgCONFIG.cfg` and the configuration sets <xx>`dcfgDCS.cfg` and <xx>`dcfgCAMERA.cfg` - see 3.2.3.1 and 3.2.3.2 - and the voltages, patterns and sequences to drive the detector).

Assuming <xxdcfg> be the name of the instrument module, install it:

```
svn co  $SVNREPO/trunk/Instruments/…/<xxdcfg>

cd <xxdcfg>/src; make all install
```

then populate the INS_ROOT:

```
ngcoDcsInstall -config <xxdcfg> [<options>]
```

2.6.1. INS_ROOT population for multi-DCS instruments

For instruments using more DCSs (see 5), `ngcoDcsInstall` must be called for each instrument detector. For instance, assuming to have four DCSs (`myCCD1`, `myCCD2`, `myCCD3`, `myCCD4`, whose configuration is defined by `xxdcfgA`, `xxdcfgB`, `xxdcfgC`, `xxdcfgD`) controlled by a Super Control process (`myINSTRUMENT`, whose configuration is defined `xxdcfgSdcs`):

```
ngcoDcsInstall -config <xxdcfgA> -instance myCCD1 [<options>]

ngcoDcsInstall -config <xxdcfgB> -instance myCCD2 [<options>]

ngcoDcsInstall -config <xxdcfgC> -instance myCCD3 [<options>]

ngcoDcsInstall -config <xxdcfgD> -instance myCCD4 [<options>]

ngcoDcsInstall -config <xxdcfgSdcs> -instance myINSTRUMENT \
           [<options>]
```

Manpage of `ngcoDcsInstall` is available in 16.1.2.

3. Startup/Shutdown Procedure

3.1. System Configuration

NGCOPT usually (see section 3.1.1) runs partly on the IWS and partly on the NGC-LLCU, where the physical interface(s) to the NGC detector front end reside (see Figure 1).

From now on, we will call `IWSENV` the online database environment which "usually" runs on the IWS and `LCUENV` the online database environment which "usually" runs on the NGC-LLCU (see section 3.1.1)

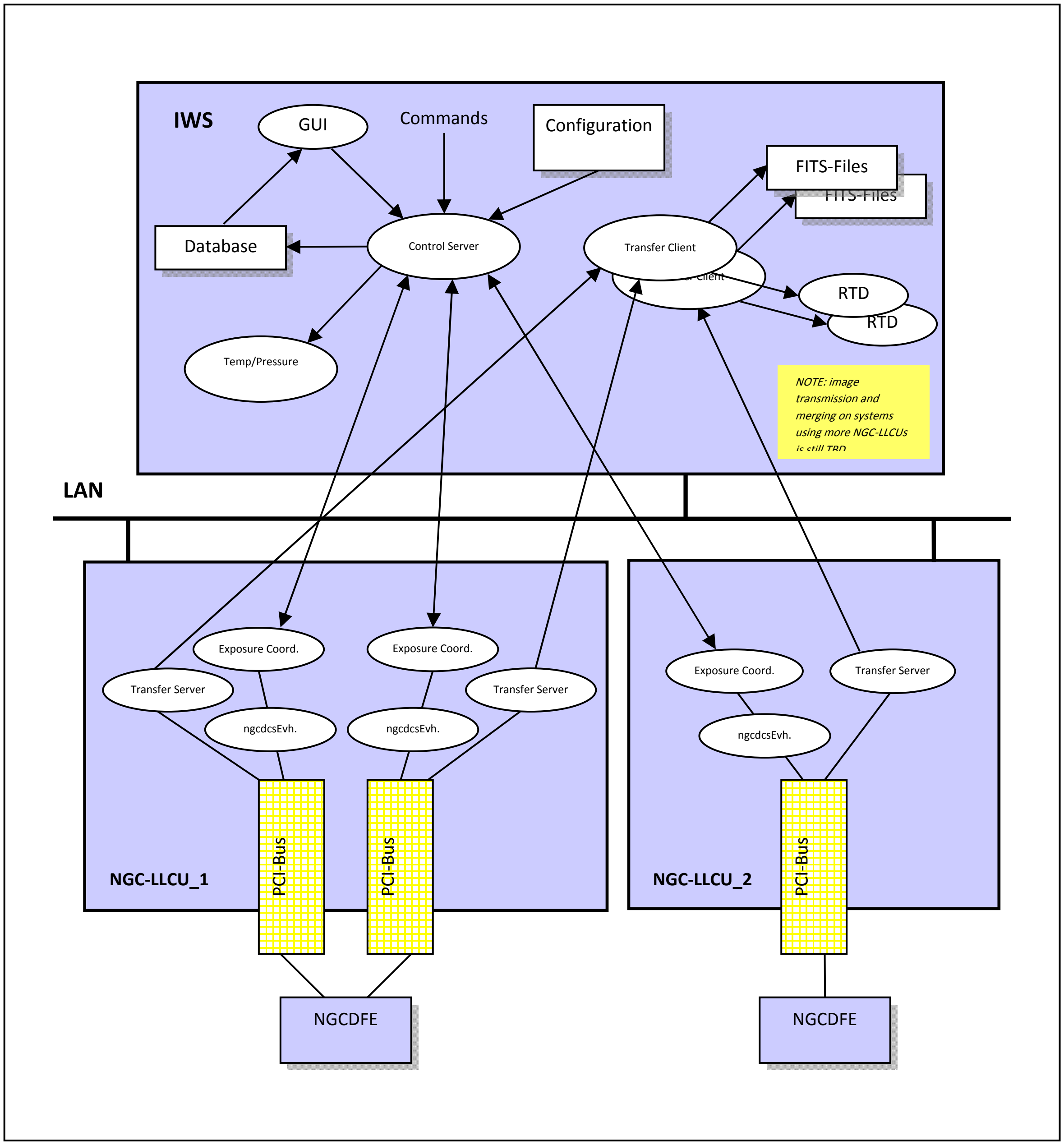


Figure 1 - Optical NGC software Architecture

For each detector system, the configuration files are kept in a separate instrument specific configuration module **`<xx>dcfg`**, which is under SVN-control. The configuration module will take care of installing all files at the proper location (i.e. `$INS_ROOT/$INS_USER/COMMON/CONFIGFILES`). In addition to the system and detector configuration file(s) there are still various other files to be maintained in such a module (e.g. voltage tables, clock pattern definitions, sequencer programs and the startup configuration as described in section 3.2.1).

3.1.1. NGCOPT operational modes

NGCOPT operates in the following different modes:

- **Normal mode**

  In Normal mode, the NGC detector electronics is connected. The NGCOPT can either be distributed on both the IWS (where the `IWSENV` online database environment is active) and the NGC-LLCU (where the `LCUENV` online database environment is active) or run completely on the NGC-LLCU (where both the `IWSENV` and the `LCUENV` online database environments are active).

- **Hardware-Simulation mode**

  In Hardware-Test mode, the NGC detector electronics is simulated (see section 3.1.3). The NGCOPT can either be distributed on both the IWS (where the `IWSENV` online database environment is active) and the NGC-LLCU (where the `LCUENV` online database environment is active) or run on a single host (where both the `IWSENV` and the `LCUENV` online database environments are active).

  This mode can be used by the higher level OS software to test the interface with the NGCOPT, when no NGC detector electronics is available.

By using the ESO VLT message system, the system configuration (i.e., where the NGCOPT processes are running) is completely transparent to the actors (instrument software, operator, engineer, etc.), because the communications between the different processes are performed through the online database environments `IWSENV` and `LCUENV`, independently from the host where these are active.

In this way, always the same software is used in all the different scenarios, in order to guarantee system robustness and behavior consistency.

NGCOPT operational mode is set by the `DET.CON.OPMODE` setup keyword in the camera configuration set (see section 3.2.3.2) or defined at startup (see section 3.2.1). Valid values are defined in `ngco.h`.

3.1.2. NGC General Purpose Control Server operational modes

NGCOPT interacts with the NGC back-end boards through the NGC General Purpose Control Server (`ngcdcsEvh`, see [AD72]). Within NGCOPT, the server operates in the following different modes:

- **Normal mode**

  In Normal mode, the NGC detector electronics is connected.

  This is the normal operational mode (default).

- **Hardware-Simulation mode**

  In Hardware-Simulation mode, the NGC detector electronics is simulated (see section 3.1.3).

- **LCU-Simulation mode**

  For NGCOPT, this mode is equivalent to Hardware-Simulation.

Server operational mode is set by the `DET.CON.DFEMODE` setup keyword in the camera configuration set (see section 3.2.3.2) or defined at startup (see section 3.2.1). Valid values are defined in `ngco.h`.

3.1.3. Simulation of the NGC detector electronics

When the NGC detector electronics is simulated, the images produced by NGCOPT contain a predefined pattern.

3.2. System Startup

3.2.1. Startup Configuration

The startup procedure is based on the common VLTSW configuration tool (“***ctoo***”, [RD75]).

For each instrument detector the installation procedure described in section 2.5 has to be performed. For each instrument detector (as well as for the Super Control process, for systems controlling more detectors, see 5), the procedure creates and installs:

- in `$INS_ROOT/$INS_USER/COMMON/CONFIGFILES/`:
  - o `<label>CONFIG.cfg`, defining the system configuration
  - o `<label>DCS.cfg`, configuration set containing the fixed physical DCS configuration (front-end electronics description, shutter, etc.)
  - o `<label>CAMERA.cfg`, configuration set containing the configurable DCS configuration (front-end electronics configuration, readout modes, etc.)
  - o `<label>/*`, directory containing all the files needed to drive the detectors (voltages, clock patterns, sequences)
- in `$INS_ROOT/$INS_USER/COMMON/SETUPFILES/DET/`:
  - o `<label>SetupComplete.det`, with the minimum SETUP needed to configure an exposure

The configuration sets describe an instance of the NGCOPT, in short FITS format, where (*) means that their usage is not yet implemented:

Keyword	Type	Description
DET.CON.INSTANCE	String	Defines the instance label for the control server and the database. Used to define the database branch and the appendix "_*<label>*" for the Control Coordination Process registered with the CCS environment. If the keyword is not present and not passed as a parameter to the startup script, the value of the $CCDNAME environment variable is used.
DET.CON.ENV	String	Defines the local online database environment under which the NGCOPT instance must run. If the keyword is not present and not passed as a parameter to the startup script, the value of the $RTAPENV environment variable is used.
DET.CON.LENV	String	Defines the remote online database environment under which the NGC-LLCU part of NGCOPT instance must run. If the keyword is not present and not passed as a parameter to the startup script, the value of the $CCDLENV environment variable is used.
DET.CON.OPMODE	String	Defines the operational mode after starting up. Valid values are "NORMAL", "HW-TEST", "HW-SIM" or "LOCAL-HW-SIM". Default is "NORMAL", in case the keyword is not present.
DET.CON.DFEMODE	String	Defines the operational mode of the NGC General Purpose Control Server after starting up. Valid values are "NORMAL", "LCU-SIM" or "HW-SIM". Default is "NORMAL", in case the keyword is not present.
DET.CON.AUTONLIN	Logical	When set to *T*, the detector system automatically goes to ONLINE at startup. Default is "F", in case the keyword is not present.
DET.CON.GUI	String	Launch graphical user interface with the specified process name. At the moment, only the default program *ngcouiPanel* is used, independently from the process name which is specified.

Keyword	Type	Description
DET.CON.DICT	String	Defines a list of dictionaries to be loaded. The common "*ESO-VLT-DIC.NGCDCS*" is always loaded into the system and needs not to be specified. The entries are separated by whit-space. Only the last descriptor of the full dictionary name is needed here (e.g. "NGCDCS STOO_CFG ...").
DET.CON.XTERM	Logical	Start all (sub-) processes in new terminal.
DET.CON.LOG (*)	Integer	Logging level. Logs system messages in the standard log-file, so that they can be seen in the CCS *logMonitor*. The level gives the detail of the messages. Default value is 0 (= no debugging, only error logging), in case the keyword is not present.
DET.CON.ITCPROC	String	Defines the name of the Image Transfer Client process which implements image post-processing. If not defined, "ngcoitc" is used.
DET.CON.OSCPROC	String	Defines the name of the Super Control process used to coordinate multi-DCS systems. If not defined, "ngcosc" is used.
DET.CON.MRGPROC	String	Defines the name of the process used to merge the image FITS files in multi-DCS systems. If not defined, the program "ngcoDcsMrgFiles" is used.

Table 1 - Startup Configuration Keywords

The startup configuration file assigns a name and some access-right attributes to the configuration set.

3.2.2. Startup Procedure

The system startup is performed through a startup script:

```
ngcoDcsStart [options]
```

The startup scripts loads the startup configuration defined by the CCDNAME environment variable or by the option –instance option, starts the coordination control process and waits - with a default timeout - until the coordination control process is active (i.e., it responds to `PING` commands).

options can be used to overwrite the values of the parameters in the configuration set keywords:

`-instance` - overwrites DET.CON.INSTANCE

`-env`	- overwrites DET.CON.ENV
`-lenv`	- overwrites DET.CON.LENV
`-opmode`	- overwrites DET.CON.OPMODE
`-dfemode`	- overwrites DET.CON.DFEMODE
`-autonlin`	- overwrites DET.CON.AUTONLIN
`-gui`	- overwrites DET.CON.GUI
`-dict`	- overwrites DET.CON.DICT (*)
`-xterm`	- overwrites DET.CON.XTERM
`-log`	- overwrites DET.CON.LOG (*)
`-itc`	- overwrites DET.CON.ITCPROC
`-osc`	- overwrites DET.CON.OSCPROC

Further special options are:

`-kill`	- kill existing NGCOPT processes, if any, before starting

3.2.3. Changes with respect to FIERA

FIERAsw configuration was online-database-driven, i.e., the configuration of a detector was described within a `.dbcfg` (database configuration) file, which was loaded at startup. NGCOPT uses *ctoo*, the `.dbcfg` file is now substituted by the startup configuration file and the configuration set.

The script `ngcoDcsStart` replaces the script `fcdDcsStart`.

If the startup script of NGCOPT is used without the options, NGCOPT will be started using the `CCDNAME`, `RTAPENV`, `CCDLENV`, `INS_ROOT`, `INS_USER`, similar to the procedure followed by the FIERASW.

3.2.4. Configuration Examples

3.2.3.1 Startup Configuration file <label>CONFIG.cfg

```
#
# Startup Configuration File
# ----------------------------------

PAF.HDR.START;
PAF.TYPE         "Configuration";   # Type of PAF
PAF.ID           "@(#) $Id: $";
PAF.NAME         "NGCOPT";          # Name of PAF
PAF.DESC         "NGCOPT Test Camera Startup Configuration";
PAF.CRTE.NAME    "ccumani";         # Name of creator
PAF.CRTE.DAYTIM  "2006-08-21";      # Civil Time for creation
PAF.LCHG.NAME    "        ";        # Name of person/appl. changing
PAF.LCHG.DAYTIM  "        ";        # Timestamp of last change
PAF.CHCK.NAME    "        ";        # Name of appl. checking
PAF.HDR.END;

#
# GENERAL CONFIG Keywords (optional)
# ----------------------------------
CONFIG.CON.LOG          T;
CONFIG.CON.BACKUP       T;
CONFIG.CON.BAKDIR       $VLTDATA/config;

#
# CAMERA CONFIG SET
# -----------------
CONFIG.SET1.NAME        "myccd";
CONFIG.SET1.DICT        "NGCDCS NGCCON";
CONFIG.SET1.FILE1       "myccdDCS.cfg";
CONFIG.SET1.PERM1       644;
CONFIG.SET1.FILE2       "myccdCAMERA.cfg";
CONFIG.SET1.PERM2       644;
CONFIG.SET1.FILE2       "myccdCAMERA_TARGET.cfg";
CONFIG.SET1.PERM2       664;

#
# ctooConfigArchive CONFIG
# ------------------------
CONFIG.ARCHIVE.NAME     "NGCOPT";
CONFIG.ARCHIVE.USER     "";
CONFIG.ARCHIVE.MODULE   "opdcfg";
CONFIG.ARCHIVE.FILE1    "opdcfg*.cfg";
CONFIG.ARCHIVE.FILE2    "opdcfg/*";

# ___oOo___
```

3.2.3.2 Configuration Set <label>DCS.cfg

```
PAF.HDR.START;                 # Start of PAF Header
PAF.TYPE         "Configuration"; # Type of PAF
PAF.ID           "";           # ID for PAF
PAF.NAME         "NGCOPT";     # Name of PAF
PAF.DESC         "NGCOPT Startup Configuration";   # Short description of PAF
PAF.CRTE.NAME    "ccumani";    # Name of creator
PAF.CRTE.DAYTIM  "2007-08-31";   # Civil Time for creation
PAF.LCHG.NAME    "        ";   # Name of person/appl. changing
PAF.LCHG.DAYTIM  "        ";   # Timestamp of last change
PAF.CHCK.NAME    "        ";   # Name of appl. checking
PAF.HDR.END;                   # End of PAF Header


####################################################################
# DEV description
####################################################################

DET.DEV1.NAME   "/dev/ngc0_com";    # associated device name


####################################################################
# CLDC description
####################################################################

DET.CLDC1.DEVIDX   1;               # associated device index
DET.CLDC1.ROUTE    "2";             # route to module
DET.CLDC1.NAME     "CLDC 1";        # optional name
DET.CLDC1.AUTOENA  F;               # auto-enable at online

DET.CLDC1.DCGN     2.0;             # bias gain
DET.CLDC1.CLKGN    1.0;             # clock gain
DET.CLDC1.TELCLKGN 1.0;             # telemetry gain
DET.CLDC1.TELDCGN  3.0;             # telemetry gain

DET.CLDC1.PREAMP   4.38;            # Preamplifier offset (volt)
DET.CLDC1.MARGIN    0.0;            # Margin for telemetry

####################################################################
# SEQ description
####################################################################

DET.SEQ1.DEVIDX    1;               # associated device index
DET.SEQ1.ROUTE     "2";             # route to module
DET.SEQ1.NAME      "Sequencer 1";   # optional name
DET.SEQ1.RUNCTRL   T;               # Run-control active
DET.SEQ1.CVTEXT    F;               # External convert active


####################################################################
# ADC description
####################################################################

DET.ADC1.DEVIDX    1;               # associated device index
DET.ADC1.ROUTE     "2";             # route to module
DET.ADC1.NAME      "ADC Module 1";  # optional name

DET.ADC1.OFFSET    1.5;             # offset value for ADC (volt)
DET.ADC1.NUM       4;               # number of ADCs on board
DET.ADC1.BITPIX   16;               # ADC bits per pixel
DET.ADC1.MON1      1;               # ADC channel to monitor

DET.ADC1.SIMMODE   0;               # simulation level of ADCs
DET.ADC1.OPMODE    0;               # operational mode of ADC-module
DET.ADC1.FIRST     T;               # first in chain
DET.ADC1.CONVERT2  F;               # convert on strobe 2
DET.ADC1.CONVERT1  T;               # convert on strobe 1
DET.ADC1.PKTCNT    0;               # packet routing length
DET.ADC1.PKTSIZE   2;               # packet size
DET.ADC1.FILTER    0;               # Filter (0 = 0.5us, 1 = 5us)
DET.ADC1.CLAMP     T;               # Analog Clamp-and-Sample
DET.ADC1.ENABLE    4;               # Number of enabled ADCs on board


####################################################################
# Preamp description
####################################################################

DET.PREAMP1.DEVIDX  1;              # Device index
DET.PREAMP1.ROUTE  "2";             # Route to module
DET.PREAMP1.NAME   "Preamp 1";      # Optional module name
DET.PREAMP1.REV    2.0;             # Preamp revision
DET.PREAMP1.BUSCMD 65;              # Delay the I2C bus by 1ms per bit


####################################################################
# SHUT description
####################################################################

DET.SHUT1.DEVIDX       1;             # Device index
DET.SHUT1.ROUTE        "2";           # Route to module
DET.SHUT1.NAME         "Shutter-1";   # Optional module name

# ___oOo___
```

3.2.3.3 Configuration Set <label>CAMERA.cfg

```
PAF.HDR.START;                 # Start of PAF Header
PAF.TYPE         "Configuration"; # Type of PAF
PAF.ID           "";   # ID for PAF
PAF.NAME         "NGCOPT";     # Name of PAF
PAF.DESC         "NGCOPT Startup Configuration";   # Short description of PAF
PAF.CRTE.NAME    "ccumani";    # Name of creator
PAF.CRTE.DAYTIM  "2007-08-31";   # Civil Time for creation
PAF.LCHG.NAME    "        ";   # Name of person/appl. changing
PAF.LCHG.DAYTIM  "        ";   # Timestamp of last change
PAF.CHCK.NAME    "        ";   # Name of appl. checking
PAF.HDR.END;                   # End of PAF Header


####################################################################
# Global info
####################################################################

DET.ID                "NGCOPT";        # Detector system Id
DET.NAME              "NGC-OPT-DCS";   # Name of detector system
DET.CHIPS             1;               # Number of chips in the mosaic
BITPIX               16;               # Number of bits per pixel

CTYPE1                "PIXEL";         # Pixel coordinate system
CRVAL1                1;               # Coordinate value of ref. pixel
CTYPE2                "PIXEL";         # Pixel coordinate system
CRVAL2                1;               # Coordinate value of ref. pixel


####################################################################
# DET configuration
####################################################################

DET.CON.INSTANCE      "myccd";         # Instance label
DET.CON.ENV           "myrtap";        # Local online database environment
DET.CON.LENV          "mylenv";        # Remote online database environment
DET.CON.OPMODE        "NORMAL";        # Operational mode
DET.CON.AUTONLIN      F;               # Go online after start
DET.CON.DICT          "NGCDCS NGCCON"; # Dictionary list
DET.CON.XTERM         F;               # Start in new terminal
DET.CON.LOG           0;               # Logging level
DET.CON.GUI           "ngcouiPanel";   # GUI Name
DET.CON.ITCPROC       "ngcoitc";       # Image Transfer Client

DET.SOFW.DETMOD       "230111"         # Instrument module version
DET.FRAM.NUMBLOCK     8;               # Default number of FITS-header blocks
DET.RTD1.REFRESH      F;               # Refresh rtd before getting a new image
DET.SEQ.TIMESIM       F;               # Enable sequencer time simulation

DET.DEV1.BOARDS       1;               # Number of boards
DET.DEV1.BOARD1.TYPE  "FEB";           # Board type
DET.DEV1.BOARD1.VERSION  "1";          # Board version (sub-type)
```

```
####################################################################
# PERIODIC WIPE
####################################################################

DET.CON.PWIPE     T;            # Autostart periodic wipe
DET.WIPE.PRGCONT  T;            # Continuous periodic wipe sequence
DET.WIPE.PRGFIL1  "pwipe.seq";  # Name of SEQi PRGFILE for wipe
DET.WIPE.CLKFIL1  "pwipe.bclk"; # Name of SEQi CLKFILE for wipe


####################################################################
# DCS configuration
####################################################################

DET.CON.SYSCFG    "ngcoptDCS.cfg"     # HW system configuration file


####################################################################
# CHIP description
####################################################################

DET.CHIP1.ID          "SER-NO 053";    # Detector chip identification
DET.CHIP1.NAME        "Marlene";       # Detector chip name
DET.CHIP1.DATE        "2006-11-22";    # Date of installation [YYYY-MM-DD]
DET.CHIP1.NX          2048;            # Physical active pixels in X
DET.CHIP1.NY          4096;            # Physical active pixels in Y
DET.CHIP1.PRSCX       50;              # Physical prescan pixels in X
DET.CHIP1.PRSCY       0;               # Physical prescan pixels in Y
DET.CHIP1.OVSCX       50;              # Physical overscan pixels in X
DET.CHIP1.OVSCY       0;               # Physical overscan pixels in Y
DET.CHIP1.PSZX        15.0;            # Size of pixel in X (mu)
DET.CHIP1.PSZY        15.0;            # Size of pixel in Y (mu)
DET.CHIP1.OUTPUTS     2;               # Number of outputs per chip

DET.CHIP1.X           1;               # X location in array
DET.CHIP1.Y           1;               # Y location in array
DET.CHIP1.XGAP        0.0;             # Gap between chips along x (mu)
DET.CHIP1.YGAP        0.0;             # Gap between chips along Y (mu)
DET.CHIP1.RGAP        0.0;             # Angle of gap between chips
DET.CHIP1.INDEX       1;               # Chip index
DET.CHIP1.LIVE        T;               # Detector alive
DET.CHIP1.TYPE        CCD;             # The Type of detector chip
DET.CHIP1.PXSPACE     1E-6;            # Pixel-Pixel Spacing

DET.CHIP1.OUT1.NAME   "NO1";           # Description of output
DET.CHIP1.OUT1.INDEX  1;               # Output index
DET.CHIP1.OUT1.ID     "IdO1";          # Output ID as from manufacturer
DET.CHIP1.OUT1.X      1;               # X location of output
DET.CHIP1.OUT1.Y      1;               # Y location of output
DET.CHIP1.OUT1.READX  -1;              # Horizontal readout direction
DET.CHIP1.OUT1.READY  -1;              # Vertical readout direction

DET.CHIP1.OUT2.NAME   "NO2";           # Description of output
DET.CHIP1.OUT2.INDEX  2;               # Output index
DET.CHIP1.OUT2.ID     "IdO2";          # Output ID as from manufacturer
DET.CHIP1.OUT2.X      2048;            # X location of output
DET.CHIP1.OUT2.Y      1;               # Y location of output
DET.CHIP1.OUT2.READX  1;               # Horizontal readout direction
DET.CHIP1.OUT2.READY  -1;              # Vertical readout direction

DET.CHIP1.CRPIX1      1;               # Ref. pixel in axis direction
DET.CHIP1.CRPIX2      1;               # Ref. pixel in axis direction


####################################################################
# MODE description
####################################################################

# MODE1

DET.MODE1.NAME        "Test1";         # Exposure mode name
DET.MODE1.DESC        "Standard";      # Exposure mode description
DET.MODE1.TRIGGER     T;               # Enable trigger
DET.MODE1.GAIN1       13;              # Gain used
DET.MODE1.GAIN2       13;              # Gain used
DET.MODE1.GAIN3       13;              # Gain used
DET.MODE1.GAIN4       13;              # Gain used
DET.MODE1.BNDWTH1     0;               # Bandwidth used
DET.MODE1.BNDWTH2     0;               # Bandwidth used
DET.MODE1.BNDWTH3     0;               # Bandwidth used
DET.MODE1.BNDWTH4     0;               # Bandwidth used
DET.MODE1.SHODEL      2000;            # [ms] Delay before opening shutter
DET.MODE1.SHDEL       2000;            # [ms] Delay before closing shutter
DET.MODE1.WREP        1;               # Wipe sequence repetition number
DET.MODE1.WCLDFIL1    "wipe1.v";       # Name of CLDCi FILE for wipe
DET.MODE1.WCLKFIL1    "mode1.bclk";    # Name of SEQi CLKFILE for wipe
DET.MODE1.WPRGFIL1    "mode1.seq";     # Name of SEQi PRGFILE for wipe
DET.MODE1.PREP        1;               # Preint sequence repetition number
DET.MODE1.DREP        1;               # Durint sequence repetition number
DET.MODE1.RREP        1;               # Readout sequence repetition number
DET.MODE1.RPRGFIL1    "read1.seq";     # Name of SEQi PRGFILE for readout
DET.MODE1.RTYPE       "Standard";      # Readout algorithm

DET.MODE1.OUTPUTS     1                # Number of outputs used for readout
DET.MODE1.ADC1.ADCS   "1";             # Outputs used for readout

DET.MODE1.OUT1.CHIP   1;               # Index of chip the output belongs to
DET.MODE1.OUT1.INDEX  1;               # Output index on the chip
DET.MODE1.OUT1.XIMA   1;               # Horizontal location of data in image
DET.MODE1.OUT1.YIMA   1;               # Vertical location of data in image
DET.MODE1.OUT1.NX     2048;            # Output data pixels in X
DET.MODE1.OUT1.NY     500;             # Output data pixels in X
DET.MODE1.OUT1.PRSCX  50;              # Output prescan pixels in X
DET.MODE1.OUT1.PRSCY  0;               # Output prescan pixels in Y
DET.MODE1.OUT1.OVSCX  50;              # Output overscan pixels in X
DET.MODE1.OUT1.OVSCY  0;               # Output overscan pixels in Y
DET.MODE1.OUT1.GAIN   0.3;             # Conversion from electrons to ADU
DET.MODE1.OUT1.CONAD  3.33;            # Conversion from ADUs to electrons
DET.MODE1.OUT1.RON    123;             # Readout noise per output (e-)

DET.MODE1.SHACTHI     F;               # Open active high
```

```
# MODE2

DET.MODE2.NAME        "Test2";         # Exposure mode name
DET.MODE2.DESC        "Seq Shutter";   # Exposure mode description
DET.MODE2.TRIGGER     F;               # Enable trigger
DET.MODE2.GAIN1       13;              # Gain used
DET.MODE2.GAIN2       13;              # Gain used
DET.MODE2.GAIN3       13;              # Gain used
DET.MODE2.GAIN4       13;              # Gain used
DET.MODE2.BNDWTH1     0;               # Bandwidth used
DET.MODE2.BNDWTH2     0;               # Bandwidth used
DET.MODE2.BNDWTH3     0;               # Bandwidth used
DET.MODE2.BNDWTH4     0;               # Bandwidth used

DET.MODE2.RREP        1;               # Readout sequence repetition number
DET.MODE2.RCLDFIL1    "acs.v";         # Name of CLDCi FILE for readout
DET.MODE2.RCLKFIL1    "acs.bclk";      # Name of SEQi CLKFILE for readout
DET.MODE2.RPRGFIL1    "acs.seq";       # Name of SEQi PRGFILE for readout
DET.MODE2.RTYPE       "FastACS";       # Readout algorithm

DET.MODE2.OUTPUTS     2                # Number of outputs used for readout
DET.MODE2.ADC1.ADCS   "1,2";           # Outputs used for readout

DET.MODE2.OUT1.CHIP   1;               # Index of chip the output belongs to
DET.MODE2.OUT1.INDEX  1;               # Output index on the chip
DET.MODE2.OUT1.XIMA   1;               # Horizontal location of data in image
DET.MODE2.OUT1.YIMA   1;               # Vertical location of data in image
DET.MODE2.OUT1.NX     1024;            # Output data pixels in X
DET.MODE2.OUT1.NY     500;             # Output data pixels in X
DET.MODE2.OUT1.PRSCX  50;              # Output prescan pixels in X
DET.MODE2.OUT1.PRSCY  0;               # Output prescan pixels in Y
DET.MODE2.OUT1.OVSCX  0;               # Output overscan pixels in X
DET.MODE2.OUT1.OVSCY  0;               # Output overscan pixels in Y
DET.MODE2.OUT1.GAIN   0.3;             # Conversion from electrons to ADU
DET.MODE2.OUT1.CONAD  3.33;            # Conversion from ADUs to electrons
DET.MODE2.OUT1.RON    100;             # Readout noise per output (e-)

DET.MODE2.OUT2.CHIP   1;               # Index of chip the output belongs to
DET.MODE2.OUT2.INDEX  2;               # Output index on the chip
DET.MODE2.OUT2.XIMA   2;               # Horizontal location of data in image
DET.MODE2.OUT2.YIMA   1;               # Vertical location of data in image
DET.MODE2.OUT2.GAIN   0.3;             # Conversion from electrons to ADU
DET.MODE2.OUT2.CONAD  3.33;            # Conversion from ADUs to electrons
DET.MODE2.OUT2.RON    200;             # Readout noise per output (e-)

DET.MODE2.SHMODE     "Sequencer1";     # Shutter operational mode


# MODE3

DET.MODE3.NAME       "Test3";          # Exposure mode name
DET.MODE3.DESC       "Digital Sample"; # Exposure mode description
DET.MODE3.TRIGGER     T;               # Enable trigger
DET.MODE3.GAIN1       13;              # Gain used
DET.MODE3.GAIN2       13;              # Gain used
DET.MODE3.GAIN3       13;              # Gain used
DET.MODE3.GAIN4       13;              # Gain used
DET.MODE3.BNDWTH1     0;               # Bandwidth used
DET.MODE3.BNDWTH2     0;               # Bandwidth used
DET.MODE3.BNDWTH3     0;               # Bandwidth used
DET.MODE3.BNDWTH4     0;               # Bandwidth used

DET.MODE3.WREP        1;               # Wipe sequence repetition number
DET.MODE3.WCLDFIL1    "wipe3.v";       # Name of CLDCi FILE for wipe
DET.MODE3.WCLKFIL1    "mode3.bclk";    # Name of SEQi CLKFILE for wipe
DET.MODE3.WPRGFIL1    "mode3.seq";     # Name of SEQi PRGFILE for wipe
DET.MODE3.PREP        1;               # Preint sequence repetition number
DET.MODE3.DREP        1;               # Durint sequence repetition number
DET.MODE3.RREP        1;               # Readout sequence repetition number
DET.MODE3.RCLDFIL1    "read3.v";       # Name of CLDCi FILE for readout
DET.MODE3.RCLKFIL1    "mode3.bclk";    # Name of SEQi CLKFILE for readout
DET.MODE3.RPRGFIL1    "read3.seq";     # Name of SEQi PRGFILE for readout
DET.MODE3.RTYPE       "SlowMosaicOffsetSimple";   # Readout algorithm

DET.MODE3.ADCSAMPL   "-1,1";           # Outputs used for readout
DET.MODE3.OUTPUTS     1                # Number of outputs used for readout
DET.MODE3.ADC1.ADCS   "2";             # Outputs used for readout

DET.MODE3.OUT1.CHIP   1;               # Index of chip the output belongs to
DET.MODE3.OUT1.INDEX  2;               # Output index on the chip
DET.MODE3.OUT1.XIMA   1;               # Horizontal location of data in image
DET.MODE3.OUT1.YIMA   1;               # Vertical location of data in image
DET.MODE3.OUT1.NX     2048;            # Output data pixels in X
DET.MODE3.OUT1.NY     500;             # Output data pixels in X
DET.MODE3.OUT1.PRSCX  50;              # Output prescan pixels in X
DET.MODE3.OUT1.PRSCY  0;               # Output prescan pixels in Y
DET.MODE3.OUT1.OVSCX  50;              # Output overscan pixels in X
DET.MODE3.OUT1.OVSCY  0;               # Output overscan pixels in Y
DET.MODE3.OUT1.GAIN   0.3;             # Conversion from electrons to ADU
DET.MODE3.OUT1.CONAD  3.33;            # Conversion from ADUs to electrons
DET.MODE3.OUT1.RON    123;             # Readout noise per output (e-)


####################################################################
# SHUT description
####################################################################

DET.SHUT1.AVAIL        T;             # Shutter available or not
DET.SHUT1.CTRL         "ngc";         # Shutter controller
DET.SHUT1.TYPE         "nostatus";    # Shutter type
DET.SHUT1.ID           "eso-01";      # Shutter unique identifier
DET.SHUT1.MODE         "Direct";      # Shutter operational mode
DET.SHUT1.OPACTHI      T;             # Open active high
DET.SHUT1.EVTEDG1      F;             # Event rising edge
DET.SHUT1.EVTEDG2      T;             # Event rising edge
DET.SHUT1.EVTEDG3      F;             # Event rising edge
DET.SHUT1.EVTEDG4      T;             # Event rising edge


# ___oOo___
```

3.2.3.4 Super Control Process Configuration Set

```
PAF.HDR.START;                 # Start of PAF Header
PAF.TYPE         "Configuration"; # Type of PAF
PAF.ID           "SUPER-DCS";  # ID for PAF
PAF.NAME         "SUPER-DCS";  # Name of PAF
PAF.DESC         "SUPER-DCS Configuration";   # Short description of PAF
PAF.CRTE.NAME    "ccumani";    # Name of creator
PAF.CRTE.DAYTIM  "2009-12-01"; # Civil Time for creation
PAF.LCHG.NAME    "        ";   # Name of person/appl. changing
PAF.LCHG.DAYTIM  "        ";   # Timestamp of last change
PAF.CHCK.NAME    "        ";   # Name of appl. checking
PAF.HDR.END;                   # End of PAF Header


# System configuration

DET.CON.INSTANCE      "myInst";            # Instance label
DET.CON.ENV           "myEnv";             # Local online database environment
DET.CON.LENV          "myLenv";            # Remote online database environment
DET.CON.OPMODE        "NORMAL";            # Operational mode
DET.CON.AUTONLIN      F;                   # Go online after start
DET.CON.GUI           "ngcouiPanel";       # GUI Name
DET.CON.DICT          "NGCDCS NGCCON";     # Dictionary list
DET.CON.XTERM         F;                   # Start in new terminal
DET.CON.LOG           0;                   # Logging level
DET.CON.ITCPROC       "";                  # Image Transfer Client
DET.CON.OSCPROC       "";                  # Super Control Process

DET.CON.INST1         "myCCD1";            # Sub-system label
DET.CON.INST2         "myCCD2";            # Sub-system label
DET.CON.INST3         "myCCD3";            # Sub-system label
DET.CON.INST4         "myCCD4";            # Sub-system label

DET.FRAM.MERGE         T;                  # Merge multiple files

DET.MODE1.NAME        "Test1";         # Exposure mode name
DET.MODE1.DESC        "Standard";      # Exposure mode description
DET.MODE1.WCLDFIL1    "wipe1.v";       # Name of CLDCi FILE for wipe
DET.MODE1.WCLKFIL1    "mode1.bclk";    # Name of SEQi CLKFILE for wipe
DET.MODE1.WPRGFIL1    "mode1.seq";     # Name of SEQi PRGFILE for wipe
DET.MODE1.PPRGFIL1    "mode1.seq";     # Name of SEQi PRGFILE for preint
DET.MODE1.DPRGFIL1    "mode1.seq";     # Name of SEQi PRGFILE for durint
DET.MODE1.RCLKFIL1    "mode1.bclk";    # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL1    "read1.seq";     # Name of SEQi PRGFILE for readout
DET.MODE1.RTYPE       "Standard";      # Readout algorithm

DET.MODE2.NAME        "Test2";         # Exposure mode name
DET.MODE2.DESC        "Seq Shutter";   # Exposure mode description
DET.MODE2.RCLDFIL1    "acs.v";         # Name of CLDCi FILE for readout
DET.MODE2.RCLKFIL1    "acs.bclk";      # Name of SEQi CLKFILE for readout
DET.MODE2.RPRGFIL1    "acs.seq";       # Name of SEQi PRGFILE for readout

DET.MODE3.NAME       "Test3";          # Exposure mode name
DET.MODE3.DESC       "Digital Sample"; # Exposure mode description
DET.MODE3.WCLDFIL1    "wipe3.v";       # Name of CLDCi FILE for wipe
DET.MODE3.WCLKFIL1    "mode3.bclk";    # Name of SEQi CLKFILE for wipe
DET.MODE3.WPRGFIL1    "mode3.seq";     # Name of SEQi PRGFILE for wipe
DET.MODE3.PPRGFIL1    "mode3.seq";     # Name of SEQi PRGFILE for preint
DET.MODE3.DPRGFIL1    "mode3.seq";     # Name of SEQi PRGFILE for durint
DET.MODE3.RCLKFIL1    "mode3.bclk";    # Name of SEQi CLKFILE for readout
DET.MODE3.RPRGFIL1    "read3.seq";     # Name of SEQi PRGFILE for readout
DET.MODE3.RCLDFIL1    "read3.v";       # Name of CLDCi FILE for readout

# ___oOo___
```

3.2.5. Editable keywords in configuration files

The configuration files described in 3.2.1 and 3.2.4 are part of the “instrument module” (see 2.5) and are not foreseen to be modified by the user.

Some keywords could be anyway be modified, if the system conditions change during operations.

These keywords are:

- `DETi.CHIPj.LIVE`

  This keyword marks a detector as alive (='T') or faulty (='F') and is used only for detector mosaics. Its default value is 'T' for all chips. If a detector is faulty, setting this keyword to 'F' marks the data in the related FITS file extension:

  ```
  HIERARCH ESO DETi CHIPj LIVE        F
  ```

- `DETi.MODEm.OUTn.GAIN`

  This keyword reports the conversion factor from electrons to ADU. During the life of the detector this value can slightly change: if this happens, this keyword should be updated. Its value is stored in the FITS header as `DETi CHIPj OUTn GAIN`

- `DETi.MODEm.OUTn.CONAD`

  This keyword reports the conversion factor from ADU to electrons. During the life of the detector this value can slightly change: if this happens, this keyword should be updated. Its value is stored in the FITS header as `DETi CHIPj OUTn CONAD`

- `DETi.MODEm.OUTn.RON`

  This keyword reports the readout noise per output (electrons). During the life of the detector this value can slightly change: if this happens, this keyword should be updated. Its value is stored in the FITS header as `DETi CHIPj OUTn RON`

- `DETi.MODEm.PREAMPn.BNDWTHi`

  This keyword reports the value for the bandwidth channel <`i`> of the preamp <n>. Usually all the preamps have the same value for channel <`i`> (and it is defined by the global keyword `DETi.MODEm.BNDWTHi`). If a single preamp needs a different value, the global setting can be overwritten by this keyword.

IMPORTANT: any modification to a configuration file must always be archived to guarantee SVN-control!

3.3. NGCOPT operational states

The NGCOPT can be in the following operational states (see [AD28]):

- **OFF**. The NGCOPT is OFF when it is not running. Consequently, the NGCOPT can never reply when it is in the OFF state.
- **LOADED.** When the NGCOPT goes to LOADED state, the database is loaded and all processes are activated. Anyway the access to hardware is not allowed.

  This is the state at the end of a successful startup.
- **STANDBY.** The software and the hardware interfaces are initialized, all hardware components are checked.

  This is the state at the end of a successful STANDBY command.

In detail all actions needed to bring the whole camera to STANDBY state are very dependent on the system hardware architecture and therefore cannot be defined in this document for all cameras. Typically the following actions are implemented:

a. Detector disconnected (voltages not applied).

b. Shutter control hardware is switched off, whenever the hardware architecture allows it.

c. Temperature monitoring remains active

d. LAN connection active (command reception enabled)

- **ONLINE**. This is the only state where the NGCOPT can perform exposures. All software and hardware is loaded, initialized and active. All voltages have been loaded. Telemetry has been acquired and checked. All the voltage switches are closed.

  This is the state at the end of a successful ONLINE command.

Figure 2 illustrates the NGCOPT operational states and the commands to switch between them (see [AD28]).

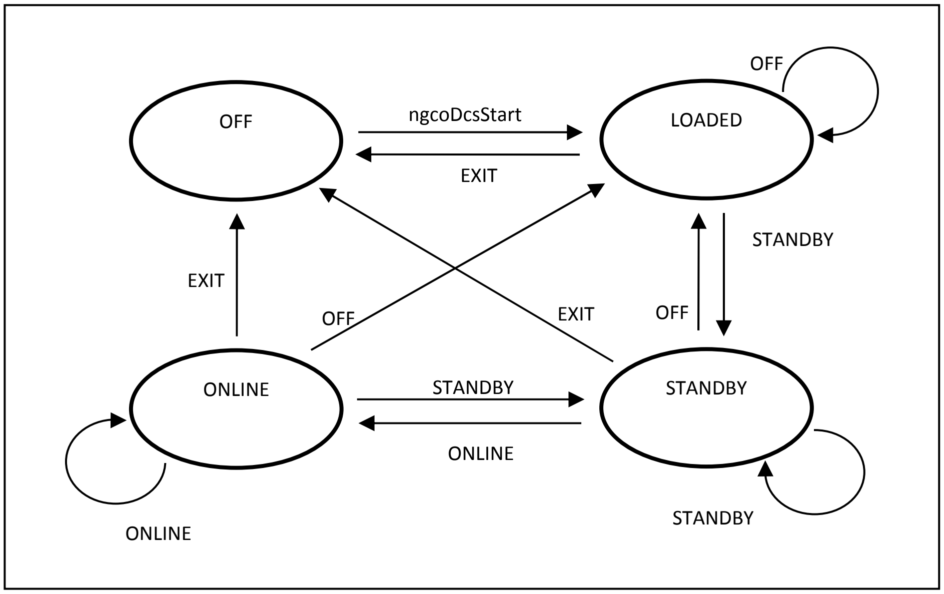


Figure 2 - Operational states and state transitions

3.3.1. Changes with respect to FIERA

NGCOPT implements the same operational states of the FIERASW.

3.4. System Shutdown

The system is shutdown by sending an `EXIT` command (see section 4) to the coordination control process `ngcocon_<label>`. The coordination control process will then shutdown all sub-processes.

A shutdown script is also available:

```
ngcoDcsStop [option]
```

The option is:

`-kill` - kill NGCOPT processes not terminated by `EXIT` command

3.4.1. Changes with respect to FIERA

NGCOPT is still shutdown by an `EXIT` command, like the FIERASW.

The script `ngcoDcsStop` replaces the script `fcdDcsStop`.

4. Command Interface

The coordination control process `ngcocon_<label>` is the only command interface between ICS and NGCOPT.

The commands which can be issued to the coordination control process are listed in the following table, where (*) means that they are not yet implemented:

Command	Parameters	Format	Description
ABORT	*none*	-	Abort running exposure.
BREAK	*none*	-	Interrupt NGC server.
CONT	-at	String <YYYY-MM-DD>T<hh:mm:ss> or "now"	Continue a paused exposure at a given time (default *now*).
DUMP (*)	*none*	-	Dump the image in memory to disk
END	*none*	-	End the current exposure(s) and read out the data.
EXIT	*none*	-	Bring the system to operational state OFF and terminate it.
INIT	*none*	-	Initialize the system. The system status goes to LOADED.
OFF	*none*	-	Bring the system to operational state LOADED.
ONLINE	*none*	-	Bring the system to operational state ONLINE.
PAUSE	-at	String <YYYY-MM-DD>T<hh:mm:ss> or "now"	Pause exposure at a given time (default *now*).
SELFTST (*)	-function	String	Execute a self-test (sw and hw) of the specified function(s).
SETUP	-file	String	Setup for the next exposure, or the running - but PAUSEd - one.
	-function	String	

Command	**Parameters**	**Format**	**Description**
STANDBY	*none*	-	Bring the system to operational state STANDBY.
START	-expoId	Integer	Exposure ID
	-at	String <YYYY-MM-DD>T<hh:mm:ss> or "now"	Start an exposure or an exposure loop (depending on the value of DET.EXP.NREP keyword, see section 7) at a given time (default *now*).
STARTWP	-periodic	Integer	Wipe chip(s) once or periodically.
STOPLP	*none*	-	Stop a loop of repeated exposures, at the end of the running exposure.
STOPWP	*none*	-	Stop a periodic wipe.
VERBOSE	-on -off	-	Set verbose mode on/off. If -*on*, the level of the logging is defined by the logging level value (which can be set/modified through the setup keyword *DET.CON.LOG*)
VERSION	*none*	-	Return current version of the NGCOPT.
WAIT	-waitMode	String "Single" / "Global"	Wait for exposure completion. Two replies are issued: one immediate with the exposure status, one at the end of the exposure.

Table 2 - Command list

Special commands which can be issued directly to the different processes are listed in the following table, where (*) means that they are not yet implemented:

Command	Parameters	Format	Description
KILL	*none*	-	Kill the process. The system status goes to OFF.
PING	*none*	-	Verify whether the process is able to send or receive messages
SIM/SIMULAT (*)	*none*	-	Put the process into simulation mode.
STATUS	*none*	-	Get the status of the process.
STOPSIM (*)	*none*	-	Stop the simulation.

Table 3 - Special command list

4.1. Changes with respect to FIERA

NGCOPT implements the same commands of the FIERASW.

To keep backward compatibility with the FIERASW as much as possible, but reducing at the same time differences with the NGC software for the infrared detectors, some command aliases have been provided (e.g., `SIM/SIMULAT`).

The command `STOPLP` replaces `STOP`.

5. Multiple Instances of DCS

For instruments controlling a single DCS. the system coordination control process ngcocon_<CCDNAME> is the only command interface between ICS and NGCOPT (see section 4).

If multiple instances of DCS are used (e.g., for instruments which control more than one NGC-LLCU), a super system coordination control process (briefly, “Super Control process”) ngcocon_<label> is the only command interface between ICS and NGCOPT.

The Super Control process is contained in the ngcosc module.

Note: a system controlling more DCSs appears to a VLT instrument as a single detector whose name is “label”. The instrument configuration file, e.g. <xx>mcfgINS.cfg, would then contain the keyword:

```
OCS.DET1.CCDNAME     "label";
```

5.1. Customization of the Super Control process

The possibility of performing customized actions at STANDBY/ONLINE/OFF/EXIT/SETUP is implemented by means of a virtual Execute method added to the corresponding action classes of the Super Control process.

The application that needs to implement a specific Super Control process must define a class that inherits from the corresponding ngcoscACTION_XXX and that overrides the Execute method. The new class must then substitute the default action manager for the event.

A template xxxosc is installed in the $VLTTOP/templates/forNGC directory and can be used to override the Super Control process.

To create a module (for instance newosc) which customizes actions of the Super Control process, go into the appropriate directory and run

```
ngcoDcsTemplate.sh -template xxxosc -module newosc
```

In the local directory a newosc module is then created.

Shortly:

- In the appropriate newosc ACTION_XXX.C file (where XXX can be any of STANDBY/ONLINE/OFF/EXIT/SETUP) implement your Execute method;
- In the newoscControl.C file, check which actions should be overridden and which should not be overridden;
- In the Makefile file, remove from newoscControl_OBJECTS the actions that you do NOT want to override.

5.2. Startup of a system using multiple instances of DCS

Proper configuration files are needed also for the startup of the Super Control process.

In particular, the `<label>CAMERA.cfg` file must contain the `DET.CON.INSTi` keywords, defining the DCSs which are used (see 3.2.3.4).

The NGCOPT program `ngcoDcsMrgFiles` performs the merging of the image FITS files produced by the different subsystems in instruments controlling multiple instances of DCS. In cases when a customized merging program is required, the name of the program must be declared in the `<label>CAMERA.cfg` file, using the `DET.CON.MRGPROC` keyword (see 3.2.1).

The `<label>CAMERA.cfg` file can also contain the `DET.FRAM.MERGE` keyword, defining if the images produced by the different DCSs should or not be merged at the end of each exposure (see 3.2.3.4). If this keyword does not exist, the default behavior is that the files are NOT merged.

5.2.1. How to run the standard Super Control process

The standard procedure described in 3.2 applies also to the startup of a system using multiple instances of DCS controlled by a standard Super Control process.

5.2.2. How to run the customized Super Control process

Assuming `newosc` to be the name of the module containing the customized Super Control process implementation, in order to use the new `newosc` process, two possibilities are offered:

1) define the `DET.CON.OSCPROC` setup keyword in the `<xx>dcfgCAMERA.cfg` configuration set:

   ```
   DET.CON.OSCPROC      “newosc”
   ```

2) launch the startup script with the `–osc <newosc>` option:

   ```
   ngcoDcsStart  –osc <newosc>
   ```

NOTE: `<newosc>` must be also the name of the IWS database branch for the Super Control process (`<myINSTRUMENT>` in the `DATABASE.db` file, see 2.3.1)

5.3. How to modify the number of instances of DCS

If a DCS instance must be removed from or added to a system using multiple instances of DCS, the following actions shall be performed:

a) Edit the Super Control Process Configuration Set (see 3.2.3.4)

   ```
   $INS_ROOT/$INS_USER/COMMON/CONFIGFILES/<myInst>CAMERA.cfg
   ```

   and remove or add the appropriate entry:

   ```
   DET.CON.INSTi
   ```

NOTE: the indexes of the defined instances must be defined in sequence, i.e.: `DET.CON.INST1`, `DET.CON.INST2`, `DET.CON.INST3`, etc. A situation like `DET.CON.INST1`, `DET.CON.INST3` (no `DET.CON.INST2`) is not correct.

NOTE: for testing purposes, it is recommended to modify only the file:

```
$INS_ROOT/$INS_USER/COMMON/CONFIGFILES/
                              <myInst>CAMERA_TARGET.cfg
```

There the `DET.CON.INSTi` entries can be modified. To remove one instance, empty the correspondent `DET.CON.INSTi` (i.e., set it to ””)

EXAMPLE:

Let’s consider the case of a system controlling four DCSs. Its file `<myInst>CAMERA.cfg` contains a configuration looking like:

```
DET.CON.INST1      "ngc1";            # Sub-system label
DET.CON.INST2      "ngc2";            # Sub-system label
DET.CON.INST3      "ngc3";            # Sub-system label
DET.CON.INST4      "ngc4";            # Sub-system label
```

To change the configuration to the case when only two DCSs are controlled, the `DET.CON.INST3` and `DET.CON.INST4` entries in the `<myInst>CAMERA.cfg` must be commented:

```
DET.CON.INST1      "ngc1";            # Sub-system label
DET.CON.INST2      "ngc2";            # Sub-system label
#DET.CON.INST3      "ngc3";            # Sub-system label
#DET.CON.INST4      "ngc4";            # Sub-system label
```

As an alternative, for a temporary solution (e.g., for testing), the following setting should be performed in the `<myInst>CAMERA_TARGET.cfg` file:

```
DET.CON.INST1      "ngc1";            # Sub-system label
DET.CON.INST2      "ngc2";            # Sub-system label
DET.CON.INST3      "";                # Sub-system label
DET.CON.INST4      "";                # Sub-system label
```

b) Edit the `$VLTDATA/ENVIRONMENTS/$RTAPENV/dbl/DATABASE.db` file and comment or uncomment/add the appropriate group of entries:

```
#define CCDNAMEi <myCCDi>
#define NGCROOTi :Appl_data:<myInst>: <myCCDi>
#define ngcdcsINSTANCEi ngcdcs_<myCCDi>
```

NOTE: the indexes of the defined instances must be defined in sequence, i.e.: `CCDNAME1`, `CCDNAME2`, `CCDNAME3`, etc. A situation like `CCDNAME1`, `CCDNAME3` (no `CCDNAME2`) is not correct.

Then regenerate the online database (see 2.3)

5.4. Shutter synchronization

See 10.4.

6. Database Interface

Some attributes of the NGCOPT online database are made public for direct read operations from external software (note: they are read only).

When accessing NGCOPT database attributes with direct CCS db calls, **applications are requested to use the macros defined in** `ngcoDbPublic.h` (see section 6.4): in this way, any change in name or location of the attribute only requires a new compilation.

All database paths below are meant to be relative to the root point for the NCG database branch.

6.1. Interface between NGCOPT and the external environment

Point	Attribute	Type	Description
system	***opMode***	dbBYTES32	Camera operational mode
System	***dfeMode***	dbBYTES32	Frontend operational mode
system	***state***	dbINT32	System operational state
system	***stateDescr***	dbBYTES32	System operational state description
exposure:config	***durintClock***	dbBYTES128	Name of the during integration clock pattern
exposure:config	***durintRep***	dbINT32	During integration sequence repetition factor
exposure:config	***durintSeq***	dbBYTES128	Name of the during integration sequence
exposure:config	***durintVolt***	dbBYTES128	Name of the during integration voltage set
exposure:config	***expMode***	dbINT32	Exposure mode index
exposure:config	***expModeDescr***	dbBYTES32	Exposure mode description
exposure:config	***expRepeat***	dbINT32	Number of exposure repetitions
exposure:config	***expTime***	dbDOUBLE	Exposure time
exposure:config	***expType***	dbBYTES32	Exposure type (Normal/Bias/Dark/etc)
exposure:config	***id***	dbINT32	Exposure identification number

<u>Point</u>	<u>Attribute</u>	<u>Type</u>	<u>Description</u>
exposure:config	***pWipeEnabled***	dbLOGICAL	Periodic wipe enabled or not
exposure:config	***pWipePeriod***	dbINT32	Wipe period
exposure:config	***preintClock***	dbBYTES128	Name of the preintegration clock pattern
exposure:config	***preintRep***	dbINT32	Preintegration sequence repetition factor
exposure:config	***preintSeq***	dbBYTES128	Name of the preintegration sequence
exposure:config	***preintVolt***	dbBYTES128	Name of the preintegration voltage set
exposure:config	***readClock***	dbBYTES128	Name of the readout clock pattern
exposure:config	***readRep***	dbINT32	Readout sequence repetition factor
exposure:config	***readSeq***	dbBYTES128	Name of the readout sequence
exposure:config	***readVolt***	dbBYTES128	Name of the readout voltage set
exposure:config	***wipeClock***	dbBYTES128	Name of the wipe clock pattern
exposure:config	***wipeRep***	dbINT32	Wipe sequence repetition factor
exposure:config	***wipeSeq***	dbBYTES128	Name of the wipe sequence
exposure:config	***wipeVolt***	dbBYTES128	Name of the wipe voltage set
exposure:control	***expLoopId***	dbUINT32	Completed exposures in loop
exposure:control	***readTime***	dbDOUBLE	Time to read image data from detector
exposure:control	***remTime***	dbDOUBLE	Remaining time to complete exposure
exposure:control	***shutter***	dbINT32	Shutter status
exposure:control	***shutterDescr***	dbBYTES32	Description of shutter status
exposure:control	***shutterSync***	dbBYTES32	Shutter synchronization type (*)
exposure:control	***state***	dbINT32	Current state of exposure
exposure:control	***stateDescr***	dbBYTES32	Description of current state of exposure

Point	Attribute	Type	Description
exposure:control	***tranPercent***	dbINT32	Percentage of image transferred to WS
exposure:control	***tranTime***	dbDOUBLE	Time to transfer image to WS
exposure:data	***binx***	dbUINT32	Horizontal binning factor
exposure:data	***biny***	dbUINT32	Vertical binning factor
exposure:data	***fileName***	dbBYTES128	Name of FITS file with image
exposure:data	***fitsHeaderBlocks***	dbUINT32	Number of blocks in FITS header
exposure:data	***fitsMtd***	dbUINT32	File saving method
exposure:data	***fitsMtdDescr***	dbBYTES128	Description of file saving method
exposure:data	***imaExtXdim***	dbUINT32 VECTOR	Vector containing the horizontal dimensions of the image extensions (*)
exposure:data	***imaExtYdim***	dbUINT32 VECTOR	Vector containing the vertical dimensions of the image extensions (*)

Table 4 - Online database attributes for detector system monitoring

(*): in a multi-DCS system, these attributes are present only in the sub-system branches.

NOTE: the interface with BOSS is still under verification, therefore the attributes above could be modified (and the definitions in `ngcoDbPublic.h` will be updated).

6.2. Interface between NGCOPT and TCS

Point	Attribute	Type	Description
wcs	***ra***	dbDOUBLE	Centre right ascension in degrees for World Coordinates display
wcs	***dec***	dbDOUBLE	Centre declination in degrees for World Coordinates display

Table 5 - Online database attributes for TCS

6.3. Image post-processing interface

The way image post-processing can be implemented is described in 9.6.

The name of the new Image Transfer Client process to be used for image post-processing can be defined in the `<xx>dcfgCAMERA.cfg` configuration set (`DET.CON.ITCPROC` keyword).

6.4. ngcoDbPublic.h

For all the above attributes, a macro is defined in the `ngcoDbPublic.h`.

When accessing NGCOPT database attributes with direct CCS db calls, applications are requested to use the macros defined in `ngcoDbPublic.h`: in this way, any change in name or location of the attribute only requires a new compilation.

6.5. Changes with respect to FIERA

NGCOPT keeps the same public online database attributes of the FIERASW.

7. Setup Command

All the parameters which are relevant for an exposure are set via a SETUP command, which must therefore be issues before starting an exposure (unless the new exposure is a perfect copy of the previous one, i.e., no parameter needs to be modified) or while an exposure is paused.

Here is a selection of the most important setup keywords (to be completed), where (*) means that their usage is not yet implemented:

Keyword	Type	Description
DET.MODE.CURID	Integer	Index of mode used for an exposure (wipe, integrate, readout)
DET.EXP.TYPE	String	Exposure type: *Normal* one integration, shutter (if any) open *Dark* one integration, shutter (if any) closed *Bias* one integration, 0 integration time, shutter (if any) closed *Flat* one integration, shutter (if any) open *Led* one integration, shutter (if any) closed, LED light source on *LedShut* one integration, shutter (if any) open, LED light source on *Multiple* DET.NDIT sub-integrations, shutter (if any) open for each integration *Burst* Multiple frames are read out, shutter (if any) always open
DET.UIT1	Double	Integration time (in seconds)
DET.BINX	Integer	Binning factor along X
DET.BINY	Integer	Binning factor along Y
DET.SHUT.ENABLE	Logical	Enable shutter control by sequencer
DET.PFLASHi.EXPTIME	Double	Preflash time (in seconds). See 9.1.7

Table 6 - Basic Setup keywords for single exposure

Keyword	Type	Description
DET.EXP.NREP	Integer	Number of repeated exposures. "0" means "forever"
DET.EXP.TIMEREP	Double	Time between two repeated exposures
DET.EXP.WIPETIM	Integer	Wipe or not before starting exposure in a loop

Table 7 - Additional Setup keywords for loops of exposures

Keyword	Type	Description
DET.NDIT (*)	Integer	Number of sub-integrations
DET.UIT<j> (*)	Double	Subintegration time (in seconds)
DET.READ.SHIFT<i> (*)	Integer	Lines shifted between integrations
DET.READ.SHIFTYP (*)	String	Line shift type: 'alternate' : 'alternate' +SHIFT1,-SHIFT1 'idem' as SHIFT1 'list' as defined in list SHIFTi

Table 8 - Setup keywords for multistep exposures

Keyword	Type	Description
DET.WIN<i>.STRX	Integer	First (lower left) window pixel in X direction
DET.WIN<i>.STRY	Integer	First (lower left) window pixel in Y direction
DET.WIN<i>.NX	Integer	Number of pixels along X
DET.WIN<i>.NY	Integer	Number of pixels along Y

Table 9 - Setup keywords for windowing

NOTE: in the actual version of NGCOPT, only single windowing is implemented, but the readout mode must support it (keyword `DET.MODEi.WINSUP=”T”`)

Keyword	Type	Description
DET.DISPLAY (*)	Integer	Real Time image display -1 no display 0 full frame display 16 bits 1 rapid frame display 16 bits
DET.CHIP<i>.CRPIX<i> (*)	Integer	Reference pixel in <i> direction.
DET.FRAM.SAMPLE (*)	Integer	Image sampling on workstation.
DET.READ.NFRAM	Integer	Defines how many sequential image data shall be stored inside a single FITS file. Default is "1"
DET.FRAM.FITSMTD	Integer	Data storage method. Possible values: 0 = 'none' (image is not saved on disk) 1 = 'compressed' 2 = 'uncompressed' 3 = 'both'
DET.FRAM.FILENAME	String	Define the base filename for the data files produced during the exposure
DET.FRAM.OFFSET	Integer	Offset (ADUs) to be applied in order to shift digitally correlated data into 16-bit range. The offset may be positive or negative to avoid either underflow or overflow. Used only by digital correlated sampling readout modes.
DET.RTD1.REFRESH	Logical	Refresh rtd before getting a new image (set rtd image to ‘0’ before getting a new image). If not defined neither in the configuration files nor via SETUP, default is “T”

Table 10 - Setup keywords for image display

Keyword	Type	Description
DET.CON.LOG	Integer	Set LOG level: 2: print shutter synchronization timestamps 1000: print all messages to BASE NGC
DET.OPT.VRELOAD	Logical	Reload the voltages in the next exposure
DET.WIPE.TIMEREP	Integer	Time interval (in seconds) for periodic wipe. If not defined neither in the configuration files nor via SETUP, default is “60”

Table 11 - Setup keywords for miscellaneous purposes

Arguments of the `SETUP` command can be file containing sets of keywords (`-file` option) or keywords (`-function` option). For example:

```
msgSend $RTAPENV ngcocon_<label> SETUP "-file mysetup.det"
msgSend $RTAPENV ngcocon_<label> SETUP "-function DET1.UIT1 2.5"
```

7.1. Changes with respect to FIERA

The setup keyword `DET.MODE.CURID` replaces `DET.READ.CLKIND`.

The setup keyword `DET.FRAM.FILENAME` replaces `DET.FRAM.FITSUNC`.

8. Status Command

The STATUS command issued to any NGCOPT process ngcocon_<label> returns the status of the process.

For debugging, the STATUS command can also be sent to the NGC general Purpose Control Server ngcdscEvh_<label> (see section 3.1.2), using the parameters described in [RD77].

8.1. Changes with respect to FIERA

The STATUS command was not implemented in the FIERASW.

9. Exposure Handling

9.1. Description

9.1.1. Exposure types

NGCOPT distinguishes among the different types of exposure defined in the Glossary (see [AD63] for a more detailed description):

- **Normal exposure** (single integration, shutter opened and closed);
- **Dark exposure** (single integration, shutter kept closed);
- **Bias exposure** (dark exposure with 0.0 s integration time);
- **Flat Field exposure** (normal exposure, chip exposed to a uniform flux of radiation);
- **Led exposure** (normal exposure, chip exposed to the radiation generated by a LED, which is located between the chip and the shutter: only the LED must be operated - see 9.1.7);
- **LedShut exposure** (normal exposure, chip exposed to the radiation generated by a LED, which is located behind the shutter: both the LED and the shutter must be operated - see 9.1.7);
- **Multiple or Multi-step exposure** (single exposure consisting of more integrations, with same or different duration. The exposure is paused after each integration. During pauses, rows may be shifted on chip);

  NOTE: this kind of exposure is not supported in the actual version of the NGCOPT**;**
- **Burst or Drift Scanning exposure** (during the integration the charges on the CCD are continuously shifted along the parallel registers and read out) **;**

  NOTE: this kind of exposure is not supported in the actual version of the NGCOPT**.**

The exposure type is defined by setting the `DET.EXP.TYPE` setup keyword (see section 7).

Accepted values for the `DET.EXP.TYPE` setup keyword are listed in macros which are defined in `ngco.h`.

9.1.2. Exposure status

When the detector system is ONLINE, an exposure can be prepared with a `SETUP` command (see section 7) and executed with a `START` command.

Schematically, starting an exposure means to:

- wipe a chip (depending on setup) : the exposure status will be *wiping*
- wait for the time to open the shutter (depending on `START` `“-at”` parameter , usually is “now”) : the exposure status will be *pending*

- open a shutter (depending on setup) : the exposure status will be *integrating*
- collect the radiation on the chip
- close a shutter (depending on setup)
- read the chip : the exposure status will be *reading*
- transfer the data to the IWS: the exposure status will be *transferring*
- merge the files received from different subsystems (only for multi-DCS systems, see 5): the exposure status will be *merging*

When the image data have been stored on disk, the exposure status goes to *completed*. If an error occurred during the exposure, the status goes to *failed*. If the exposure was aborted, the status goes to *aborted*.

Generally the field of view can already be changed (e.g. telescope can be moved) when the exposure status changes to *transferring* (all data for this exposure have been read-out).

By default, with NGCOPT it is possible to start an exposure only when one of the completion states (*success*, *failure*, *aborted*) have been reached (i.e., after the image data produced by the previous exposure have been saved on disk).

If the time between end of detector readout and availability of the FITS file on disk becomes a significant overhead, NGCOPT can be instructed to start an exposure right after the end of the transmission of the image date of the previous exposure to the IWS, by using the setup keyword TBD.

The current exposure status value is stored in the database attribute

```
<alias>${CCDNAME}:exposure:control.state
```

The value of the current exposure status can be:

INACTIVE
PENDING
WIPING
INTEGRATING
PAUSED (i.e., shutter temporary closed)
READING
PROCESSING (i.e., processing image data, if requested by SETUP)
TRANSFERRING (i.e., transferring image data to IWS)
MERGING (merge subsystem files: only for multi-DCS systems, see 5)
COMPLETED (i.e., completed successfully)
FAILED (i.e., completed with error)
ABORTED (i.e., completed without data readout, on request)

Macros for the exposure status values and descriptions are defined in `ngco.h`.

9.1.3. Image data

Image data are provided by NGCOPT in two ways:

- Raw-data for Real-time display (see section 13)
- FITS files (see section 9.3)

Whenever a new data file is created, the full path name is written into the database attribute

`<alias>${CCDNAME}:exposure:data.fileName`

9.1.4. Exposure Id

In order to be able to uniquely identify an exposure, an identification number (exposure Id) is associated to each exposure.

The exposure Id should be passed to the NGCOPT as a parameter of the command START (as defined in [AD28]). If the command `START` has no exposure Id parameter, the exposure Id is defined by NGCOPT.

The exposure Id is returned as a reply parameter to the command `START`.

9.1.5. Readout types

Different readout algorithms are assigned to the exposure modes by the `DET.MODEi.RTYPE` keywords in the `<label>CAMERA.cfg` file (see 3.2.3.3):

- “`Standard`” is a general purpose readout algorithm, supporting digital multisampling. If the `DET.MODEi.RTYPE` keyword is not defined in the `<label>CAMERA.cfg` file, this (default) readout algorithm is used;
- “`FastACS`” is optimized for fast readout rates using analog clamp&sample;
- “`SlowMosaic`” is for readout of big detectors, like mosaics, supporting digital multisampling;
- “`FastMosaic`” is optimized for fast readout rates of big detectors using analog clamp&sample;
- “`SlowMosaicOffset`” is for digital correlated sampling performed on weighted values;
- “`SlowMosaicOffsetSimple`” is for digital correlated sampling performed on rough values.

9.1.6. Shutter types and modes

Different shutter types are defined by the `DET.SHUTi.TYPE` keyword in the `<label>CAMERA.cfg` file (see 3.2.3.3):

- "nostatus" is for shutter without status signals;
- "Bonn" is for Bonn-type shutters, with status signals (see [RD8])

A shutter can be operated in different modes (see [RD8]):

- "Normal": the shutter is automatically controlled by the hardware

  NOTE: this mode should never be used with NGCOPT
- "Direct": the shutter is controlled by the software interfacing the NGC shutter control module;
- "Sequencer1": shutter is controlled by a clock bit from the sequencer and the shutter status is ignored;
- "Sequencer2": Shutter is controlled by the sequencer but shutter module handshakes to sequencer.

NOTE: if the shutter is operated in "Sequencer1" or "Sequencer2" mode, the opening and closing of the shutter is performed by the sequencer, usually within the "read" sequence (defined by the keyword).

Default settings for the shutter mode are defined via the DET.SHUTi.MODE keywords in the <label>CAMERA.cfg file (see 3.2.3.3).

Different shutter modes per exposure mode can be set by the DET.MODEi.SHMODE keywords in the same <label>CAMERA.cfg file (see 3.2.3.3):

9.1.7. External LEDs

External LEDs can be controlled via the shutter port on the Transition Board of the NGC Front-End Electronics.

Different "shutter" types must defined by the DET.SHUTi.TYPE keyword in the <label>CAMERA.cfg file (see 3.2.3.3):

- "FlatLed" is for LEDs used for Flat Fields, i.e., turned on during the integration (in "Led" and "LedShut" exposure types, see 9.1.1);
- "PreflashLed" is for LEDs used for preflashing the detectors, i.e., turned on after the wiping and before the opening of the beginning of the integration. The duration of the preflashing is defined by the DET.PFLASHi.EXPTIME - see 7 - setup keywords (if more preflash LEDs are used, they can be on for different times, but the shutter will be open only at the end of the longest preflash exposure time)

9.1.8. Time stamps

Depending on the shutter mode and the exposure setting, the "start exposure" time stamps (keywords `MJD-OBS` and `DATE-OBS` in the FITS file header) label a different exposure step:

a. If shutter operated in "`Direct`" mode:

- In case of a DARK or BIAS, the "start exposure" time stamps report the time of the end of the wipe
- In case of a NORMAL exposure, the "start exposure" time stamps report the time of the starting of the integration time (i.e., when the shutter has been completely open)

b. If shutter operated in "`Sequencer1`" or "`Sequencer2`" mode:

- the "start exposure" time stamps report the time of the start of the readout sequence

"Start exposure" accuracy has been measured to be better than 6 ms.

9.1.9. Changes with respect to FIERA

NGCOPT implements the exposure and status types of the FIERASW. In addition, the MERGING exposure state has been added for multi-DCS systems (see 5).

NGCOPT defines the same exposure status numerical values of the FIERASW, a part from the one labeling the WIPING status (see `ngco.h`).

NOT YET IMPLEMENTED: Has a new feature, it will be possible to start a new exposure when the data of the previous one have been transmitted to the IWS, but not stored on disk yet (although this will NOT be the default behavior).

9.2. Commands

Exposures are prepared using the **`SETUP`** command and started using the **`START`** command.

A timed exposure start can be done using the **-at** option:

```
START –at <YYYY-MM-DD>T<hh:mm:ss>
```

The value of the -at parameter defines an absolute time (UTC) for the opening of the shutter (an absolute time for a dark exposure has no sense). Until the actual start time is reached, the exposure status is set to "pending", which will limit the set of accepted commands during that time.

An exposure can be paused using the command **`PAUSE`** (note that the time the exposure is PAUSE'd will be added to the dark's time, see [RD63]). The shutter is closed and the counting of the remaining exposure time suspended. The exposure is then restarted by the command **`CONTINUE`**.

The exposure can be aborted using the command **`ABORT`**. In this case no data file is generated unless a frame was already received at the time when the command was issued.

The command **`END`** makes the acquisition process terminate the exposure as soon as possible and generate a data file.

The command **`WAIT`** can be used to wait for an exposure to complete. A reply message with the current exposure status is sent immediately. When the exposure status is (or becomes) “completed” (i.e. “success”, “failure” or “aborted”), NGCOPT sends the last reply, which again contains the actual exposure status. A running exposure always has to be waited for completion before starting the next one or before issuing a new setup.

WARNING: in the actual version of NGCOPT, the `WAIT` command returns when the image transfer has been completed (`SETUP` for the next exposure can be sent), and the completion of the image FITS file storing is shown by the updating of the `exposure:data.fileName` database attribute.

It must be clarified if this will be the final behavior, or if the `WAIT` command should return after the image storage has been completed.

Typical command sequences are:

```
a) START – WAIT
b) START – PAUSE - CONTINUE – WAIT
c) START – END – WAIT
d) START - ABORT – WAIT
```

Alternatively, the exposure status attribute in the database (see section 6.1) may be used to wait for a specific state (e.g. transferring).

9.2.1. Changes with respect to FIERA

NGCOPT implements the same exposure commands of the FIERASW.

9.3. File Formats

If data storage is enabled, images are saved in the `$INS_ROOT/$INS_USER/DETDATA` directory as FITS files compliant with [RD37], i.e., using the “image extension per chip" format. In this format, data are ordered by chip: each CCD corresponds to an extension. A primary header sits on the top of the file.

To enable "data cubes" (i.e., saving n successive frames into a single FITS file) the setup parameter `DET.READ.NFRAM` must be set to a value different from 1 (see section 7). All the frames related to the same chip are stored in the same image extension.

Currently the formats supported for pixels values are 16-bits and 32-bits.

Independently from the readout mode used, the complete physical image is stored in one single FITS file per detector. In case of multi-DCS instruments (see 5) merging can be enabled to provide one single FITS file for the whole instrument. Multiple windows are also stored in different IMAGE extensions of a single FITS file.

9.3.1. Changes with respect to FIERA

NGCOPT implements the same file format of the FIERASW.

9.4. Naming Schemes

FITS file names are defined by the setup keyword `DET.FRAM.FILENAME` (see section 7).

In case the number of FITS file to be produced is more than one (`DET.EXP.NREP` setup parameter: see section 7), NGCOPT assumes that all files will have the same name, followed by a sequential integer index, starting from 0.

Example: if `DET.EXP.NREP` is set to 3 and `DET.FRAM.FILENAME` is set to *`myImage.fits`*, NGCOPT will look for files *`myImage.fits`* (first exposure), *`myImage.1.fits`* (second exposure) and *`myImage.2.fits`* (third exposure)

9.4.1. Changes with respect to FIERA

NGCOPT implements the same naming scheme of the FIERASW.

9.5. FITS-Header Contents

NGC FITS keywords are defined in the `ESO-VLT-DIC.NGCDCS` dictionary (***dicNGC*** SVN module).

Apart from the image raw data, NGCOPT is also responsible for providing keywords for the FITS header. Depending on their type, keywords are treated in two different ways.

- **Standard keywords**. Some basic keywords, needed by any image analysis system to read the FITS file, are written at the beginning of the file.
- **Hierarchical keywords**. They are not strictly needed to interpret the pixel values and normally do not appear at the beginning of the FITS header.

9.5.1. Changes with respect to FIERA

NGCOPT implements the same FITS structure of the FIERASW.

No more separate `.det` files are created: NGCOPT writes all the information directly in the FITS file.

9.6. Image post-processing

The possibility of performing customized post processing is implemented by means of a virtual `PostProcessing` method added to the `DATA_IN` action class of the Image Transfer Client process (module `ngcoitc`).

The application that needs to implement some specific post processing must define a class that inherits from `ngcoitcACTION_DATA_IN` and that overrides the `PostProcessing` method. The new class must then substitute the default action manager for the `DATA_IN` event.

Among procedures that can be implemented in this way are e.g. centroiding and bias subtraction.

Figure 3 and Figure 4 describe the class and the sequence diagrams of the mechanism.

9.6.1. Creating a module which implements post-processing

Instructions and a template are provided in the `xxxoitc` module which is installed in the `$VLTTOP/templates/forNGC` directory.

To create a module (for instance `newoitc`) which implements post-processing, go into the appropriate directory and run

```
    ngcoDcsTemplate.sh -template xxxoitc -module newoitc
```

In the local directory a newoitc module is then created.

By running:

```
    cd newoitc/src ; make man
```

the documentation concerning the implementation of post-processing is the installed in the `newoitc/doc` directory (in MagicDrawXML and html format)

Shortly:

- In the `newoitc ACTION_MYDATA_IN.C` file, implement your PostProcessing method;
- In the `newoitcControl.C` file, check which actions should be overridden and which should not be overridden (e.g., do you need or not to override the `SETUP` command?);
- In the `Makefile` file, remove from `newoitcControl_OBJECTS` the actions that you do NOT want to override (e.g, `SETUP`).

9.6.2. How to run the process which implements post-processing

Assuming newoitc to be the name of the module containing the post-processing implementation, in order to use the new newoitc process, two possibilities are offered:

3) define the `DET.CON.ITCPROC` setup keyword in the <xx>`dcfgCAMERA.cfg` configuration set:

```
DET.CON.ITCPROC     “newoitc”
```

4) launch the startup script with the `–itc <newoitc>` option:

```
ngcoDcsStart  –itc <newoitc>
```

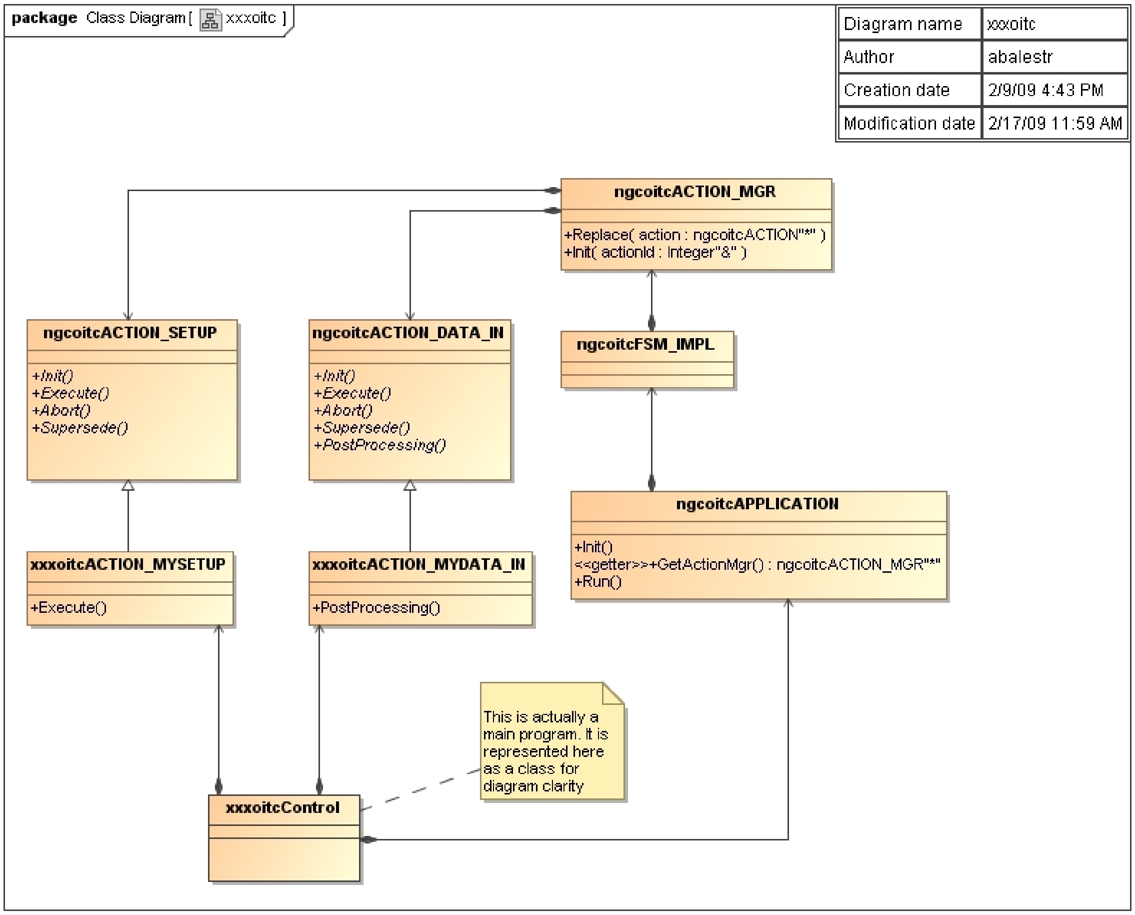


Diagram name	xxxoitc
Author	abalestr
Creation date	2/9/09 4:43 PM
Modification date	2/17/09 11:59 AM

Figure 3 - Post Processing: Image Transfer Client class

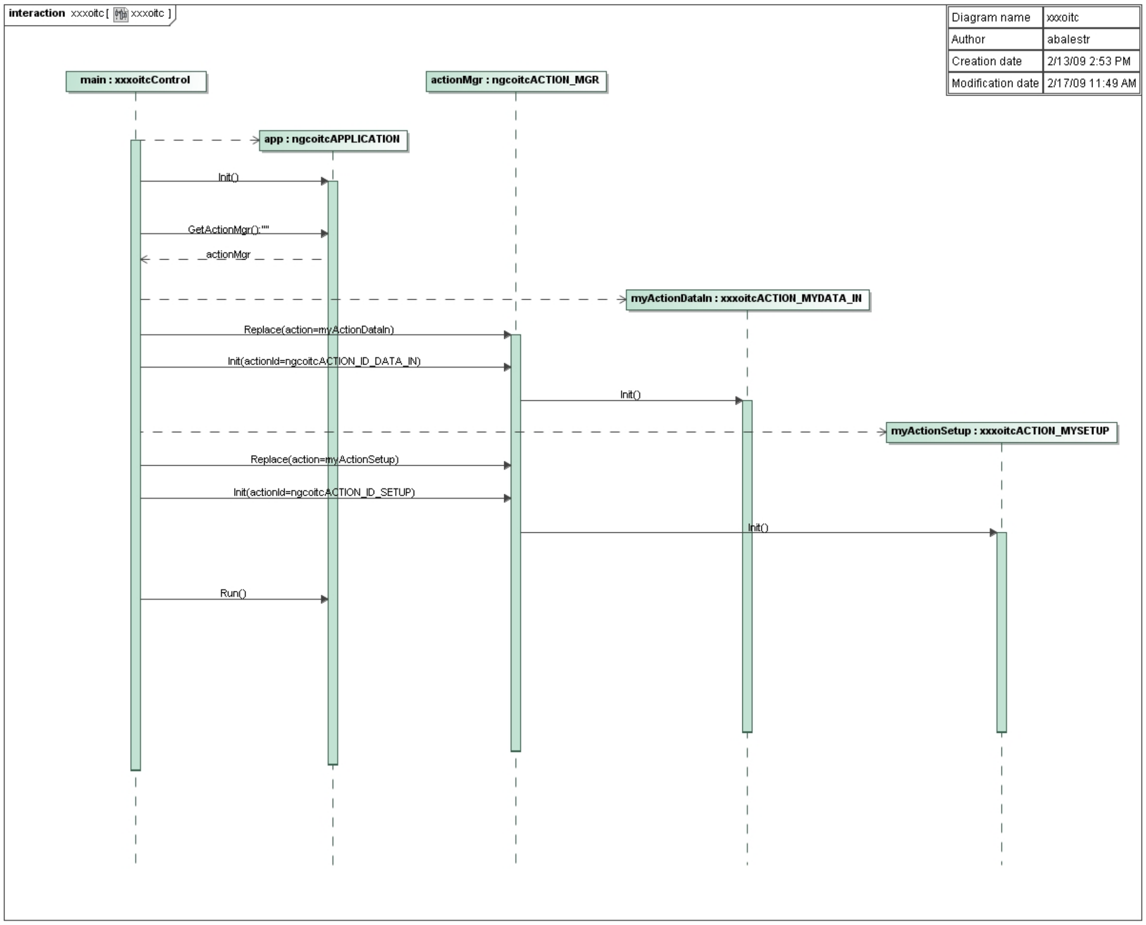

interaction xxxoitc [xxxoitc]
Diagram name xxxoitc
Author abalestr
Creation date 2/13/09 2:53 PM
Modification date 2/17/09 11:49 AM
main : xxxoitcControl
actionMgr : ngcoitcACTION_MGR
app : ngcoitcAPPLICATION
Init()
GetActionMgr():""
actionMgr
myActionDataIn : xxxoitcACTION_MYDATA_IN
Replace(action=myActionDataIn)
Init(actionId=ngcoitcACTION_ID_DATA_IN)
Init()
myActionSetup : xxxoitcACTION_MYSETUP
Replace(action=myActionSetup)
Init(actionId=ngcoitcACTION_ID_SETUP)
Init()
Run()

Figure 4 - Post Processing: Image Transfer Client sequence diagram

10. Synchronization

10.1. Synchronization via external trigger

Synchronization points can be inserted at any place in any clock pattern executed by the sequencer program (i.e. set the "*wait-for-trigger*" bit in the particular state). When reaching such a point, the pattern execution is suspended until the arrival of an external trigger signal (see [RD9] and [RD8] for signal timing and accuracy). Via this external trigger input it is possible to synchronize exposures on multiple NGCOPT instances. The external trigger signal is also used to synchronize detector read-outs with external devices. Using the VLT-TIM for generating the trigger pulse(s), synchronization at absolute times is possible. Some signal lines are available to in turn trigger external devices (e.g. tell another device, that a read-out has finished).

10.2. Synchronization via global run-signal

If several sequencers are installed in the same system (i.e. the same instance of NGCOPT), then the exposure start can be synchronized by using the global run-signal, which is raised by one sequencer instance and is propagated to all other sequencer instances having the external run-control enabled ("`DET.SEQi.RUNCTRL = T`" in the detector configuration file, see [RD77]).

10.3. Synchronization via system time

If no high accuracy is needed, the synchronization can be also done at command interface level (e.g. issue an exposure start command at the proper time or use the command "`START –at <start-time>`").

10.4. Synchronization with shutter operations

If a system controlling multiple instances of DCS (see 5) uses only one shutter, the integration and readout on the different DCS sub-systems must be synchronized with the shutter operations. This is performed internally by the NGCOPT. To achieve this, the system controlling the shutter must be defined as "Shutter-Master", by setting the following keyword in the `<label>CAMERA.cfg` configuration file:

```
DET.SHUTi.SYNC    "Master";      # Shutter sync mode
```

The other subsystem must be defined as "Shutter-Slave", by setting the following keyword in the `<label>CAMERA.cfg` configuration file:

```
DET.SHUTi.SYNC    "Slave";       # Shutter sync mode
```

Synchronization is then performed as follows:

a. at the beginning of the exposure, the "Master" system opens the shutter when all the "Slave" systems have completed the wipe and have run the sequence setting the detector in integration (the so called "prepare integration" sequence);

b. at the end of the exposure, the "Slave" systems start the readout only when the shutter has been completely closed.

In addition, a subsystem can be instructed to wait before starting the readout by an amount of milliseconds which is defined - per each mode - in the `<label>CAMERA.cfg` configuration file by the following keyword:

```
  DET.MODEi.SHDEL   <ms>;  # ms delay btw shutter close and readout
```

NOTE: An error in the `DET.SHUTi.SYNC` setting (for instance, if this keyword is set, but no other DCS subsystems are operated, or if all DCS-subsystems have this keyword set to the same value) causes a timeout error while running an exposure (“Master” system waiting for all “Slave” system to have set the system in integration, or “Slave” system waiting for the “Master” to have close the shutter).

11. Error Definitions

The CCS error mechanism [RD32] provides a classification scheme for application specific errors.

The meaning of the error class and the possibly needed interactions are described in a help file (.hlp), which can be displayed with the standard CCS-tools (also with the logMonitor).

The detailed error reason (e.g., command which failed, wrong parameter issued and boundary values, etc) is given in an associated error message string.

NGCOPT uses the errors defined by ngcb (see [RD77]).

In addition, NGCOPT modules define their own errors. The errors which are common to all the modules are listed in the following table:

Error	Severity	Description
ngco<mod>ERR_FATAL	fatal	Fatal internal error: %.40s
ngco<mod>ERR_CREATE	serious	Failure creating: %.40s
ngco<mod>ERR_INIT	serious	Failure initializing: %.40s
ngco<mod>ERR_FUNCTION	serious	Failure invoking function: %.40s
ngco<mod>ERR_NULL_POINTER	warning	Pointer to %.40s is NULL
ngco<mod>ERR_ZERO	warning	Division by zero while %.20s
ngco<mod>ERR_NOT_FOUND	warning	Object %.20s (ID: %d) not found
ngco<mod>ERR_DB_READ	serious	Failed to read from DB %.35s %.35s%.35s
ngco<mod>ERR_DB_WRITE	serious	Failed to write to DB %.35s %.35s%.35s
ngco<mod>ERR_SEND_COMMAND	serious	Failed to send command %.20s to %.40s
ngco<mod>ERR_SEND_REPLY	serious	Failed to send %.20s reply for command %.20s to %.40s
ngco<mod>ERR_REPLY	serious	Error reply to command %.20s received from %.40s

Error	Severity	Description
ngco<mod>ERR_TIMEOUT	serious	Command %.20s sent to %.40s timed out
ngco<mod>ERR_PARAMETER	warning	Parameter %.20s has invalid value
ngco<mod>ERR_ABORTED	warning	Command %.20s aborted
ngco<mod>ERR_STATE	serious	Command %.20s not allowed in state %.40s
ngco<mod>ERR_ACTION	serious	Action %.20s (%.15s) failed in state %.40s
ngco<mod>ERR_EVENT	serious	Event %.20s not handled in state %.40s
ngco<mod>ERR_ASSERT	serious	Assertion Failed %.20s %.40s %d.

Table 12 - Errors common to all NGCOPT modules

The errors which are specific to certain modules of the NGCOPT are listed in the following tables:

Error	Severity	Description
ngcoexpERR_BRD_TYPE	fatal	Wrong board%d type: %.20s instead of %.20s
ngcoexpERR_BRD_VERSION	fatal	Wrong board%d version: %.20s instead of %d
ngcoexpERR_CANNOTSKIPWIPE	warning	No preintegration sequence: cannot skip wipe
ngcoexpERR_DCS	fatal	Error received from ngcdcsEvh to command %.80s
ngcoexpERR_DMA_ALLOC	fatal	Error allocating dma ring-buffers: %.80s
ngcoexpERR_DMA_CLOSE	warning	Error closing dma: %80s
ngcoexpERR_DMA_INIT	fatal	Error initialising dma: %.80s

Error	Severity	Description
ngcoexpERR_DMA_OPEN	fatal	Error opening dma: %.80s
ngcoexpERR_DMA_OVERRUN	fatal	Dma ring-buffer overrun: %.80s
ngcoexpERR_DMA_STOP	serious	Error stopping dma: %.80s
ngcoexpERR_DMA_TIMEOUT	fatal	Timeout while waiting for dma data: %.80s
ngcoctrERR_END	warning	END command accepted only when exposure is INTEGRATING, PAUSED or READING
ngcoexpERR_FILE_SIZE	serious	Generated file too big, size is %dx%d
ngcoexpERR_FILE_SIZENFRAM	serious	Generated file too big, reduce NFRAM (now %d) below the max value for this mode (%d)
ngcoexpERR_HD_SPACE	serious	Not enough hard disk space to store image (needed %d, available %d)
ngcoexpERR_HDU_KEYWORDS	serious	Too many keywords: %d (max %d)
ngcoexpERR_IRKEY	warning	keyword %.80s is for IR systems
ngcoexpERR_MERGESYNC	serious	Timeout waiting for image merging completion
ngcoexpERR_MULTISTEP	serious	No sequence defined for Multistep Exposure
ngcoconERR_NOROUTE	warning	Command %.20s not found in routing table
ngcoexpERR_NOSHUTTER	seriuos	No shutter is available
ngcoitcERR_NOT_DEFINED	fatal	Variable %.20s not defined
ngcoexpERR_PAUSE	warning	Dark exposure cannot be paused
ngcoexpERR_PERIODIC_WIPE	warning	Could not load and start sequence for wipe

Error	Severity	Description
ngcoexpERR_READSYNC	serious	Timeout waiting for image transfer completion
ngcoscERR_REMOVE	warning	Error removing file %.40s
ngcoitcERR_RTD	serious	Error communicating to RTD %.40s
ngcoexpERR_SEM	fatal	Error handling semaphore: %.80s
ngcoexpERR_SEM_LOCK	fatal	Cannot lock semaphore %d %.80s
ngcoexpERR_SEM_UNLOCK	fatal	Cannot unlock semaphore %d %.80s
ngcoexpERR_SETUP_UNDEF	serious	SETUP keyword %.80s undefined
ngcoexpERR_SETUP_WRONGINT	fatal	SETUP keyword %.80s value %d is out of range
ngcoexpERR_SETUP_WRONGSTR	fatal	SETUP keyword %.80s has unacceptable value %.80s
ngcoexpERR_SHUTDELAY	serious	Error reading shutter %.16s delay
ngcoexpERR_SHUTTER	fatal	Shutter error status bit set
ngcoexpERR_SHUTSYNC	warning	Timeout waiting for shutter synchronisation flag be %d
ngcoitcERR_SYSTEM	fatal	Function %.40s failed: %40s
ngcoexpERR_TIME_FORMAT	serious	Wrong format for start-at time (%.80s)
ngcoexpERR_UK_SEQ_STATUS	fatal	Unknown sequencer status: %.80s
ngcoexpERR_WIPE_TIMER	seriuos	Could not handle timer for periodic wipe
ngcoexpERR_WIPE_SEQ	serious	No sequence defined for periodic wipe
ngcoexpERR_WR_SEQ_STATUS	fatal	Wrong sequencer status: %.80s (instead of %.80s)

Table 13 - Errors specific to the different NGCOPT modules

NOTE: it is under investigation the usage of an error definition file common to all the modules.

12. Error and Logging Handling

Error and system logging is performed using the standard CCS error and logging systems (see [RD32]).

NOT YET IMPLEMENTED: Additionally the verbose output can be logged in a detail depending on the given log-level (see setup keyword `DET.CON.LOG` in section 3.2.1 and command `VERBOSE` in section 4) for maintenance and debugging purposes.

Operational logs are TBD.

13. Real-Time Display Interface

NGCOPT provides **raw data** for the VLTSW real-time display utility `rtd` (for VLTSW versions up to `VLT2010`) or `rtdcore` (from `VLT2011` on).

It is possible to modify the default real-time display utility by setting the environment variable `NGCO_RTD` to `rtd` or `rtdcore` **before compiling** and running the code.

The mechanism to deliver raw data is the same as defined in [RD40].

Raw-data are written in shared memory as they come out from the Detector Electronics, namely with full resolution (16 or 32-bits unsigned integer). No reduction (e.g. to 8-bits) is done by NGCOPT.

In addition to the display of the raw-data, NGCOPT supports also the display of **World Coordinates** through `rtd` or `rtdcore`. One point in the NGC branch of the online database is dedicated to this feature (see section 6.2).

13.1. Changes with respect to FIERA

Till `VLT2010`, NGCOPT provides the same interface to `rtd` of the FIERASW.

From VLT2011, NGCOPT provide interface to `rtdcore`.

14. Graphical User Interface

14.1. Control Panels

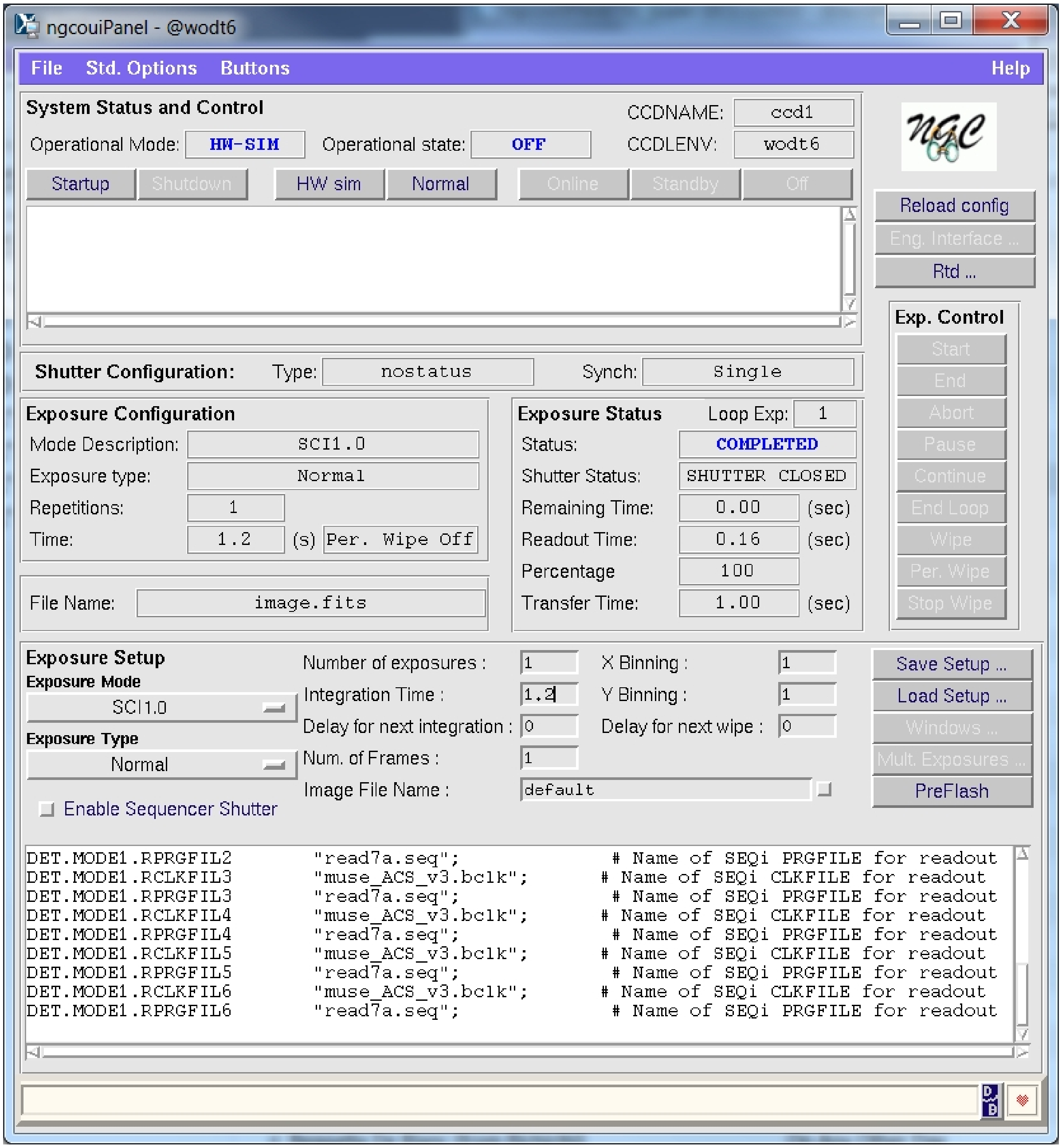
A graphical user interface is provided to operate NGCOPT in standalone mode.

One single panel, shown in Figure 5, provides all functionality needed to startup/shutdown the NGC software, define an exposure setup, start and control an exposure, display an image as result of an exposure.

The same panel is used, independently if and which parts of the CCD system used are simulated.

To startup the optical NGC Control Panel run

```
    ngcouiPanel &
```



System Status and Control

Operational Mode: HW-SIM
Operational state: OFF
CCDNAME: ccd1
CCDLENV: wodt6

Shutter Configuration: Type: nostatus Synch: Single

Exposure Configuration

Mode Description: SCI1.0
Exposure type: Normal
Repetitions: 1
Time: 1.2 (s) Per. Wipe Off

File Name: image.fits

Exposure Status Loop Exp: 1

Status: COMPLETED
Shutter Status: SHUTTER CLOSED
Remaining Time: 0.00 (sec)
Readout Time: 0.16 (sec)
Percentage 100
Transfer Time: 1.00 (sec)

Exposure Setup

Exposure Mode: SCI1.0

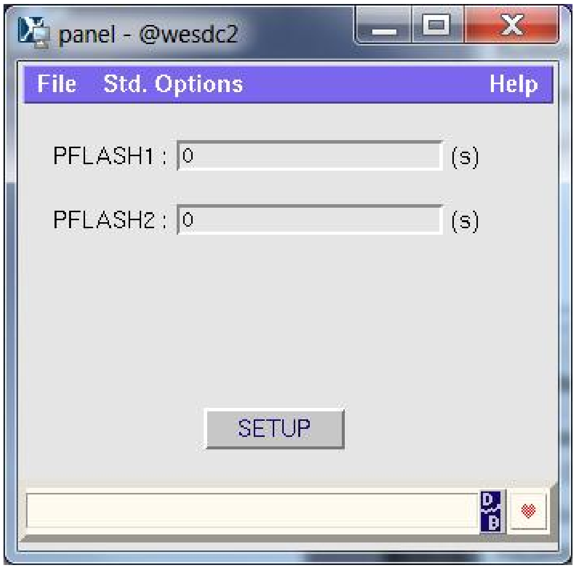
Exposure Type: Normal

□ Enable Sequencer Shutter

Number of exposures : 1
Integration Time : 1.2
Delay for next integration : 0
Num. of Frames : 1
Image File Name : default □
X Binning : 1
Y Binning : 1
Delay for next wipe : 0

```
DET.MODE1.RPRGFIL2        "read7a.seq";            # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL3        "muse_ACS_v3.bclk";     # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL3        "read7a.seq";            # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL4        "muse_ACS_v3.bclk";     # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL4        "read7a.seq";            # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL5        "muse_ACS_v3.bclk";     # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL5        "read7a.seq";            # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL6        "muse_ACS_v3.bclk";     # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL6        "read7a.seq";            # Name of SEQi PRGFILE for readout
```

Figure 5 - Optical NGC Control Panel



PFLASH1 : 0 (s)

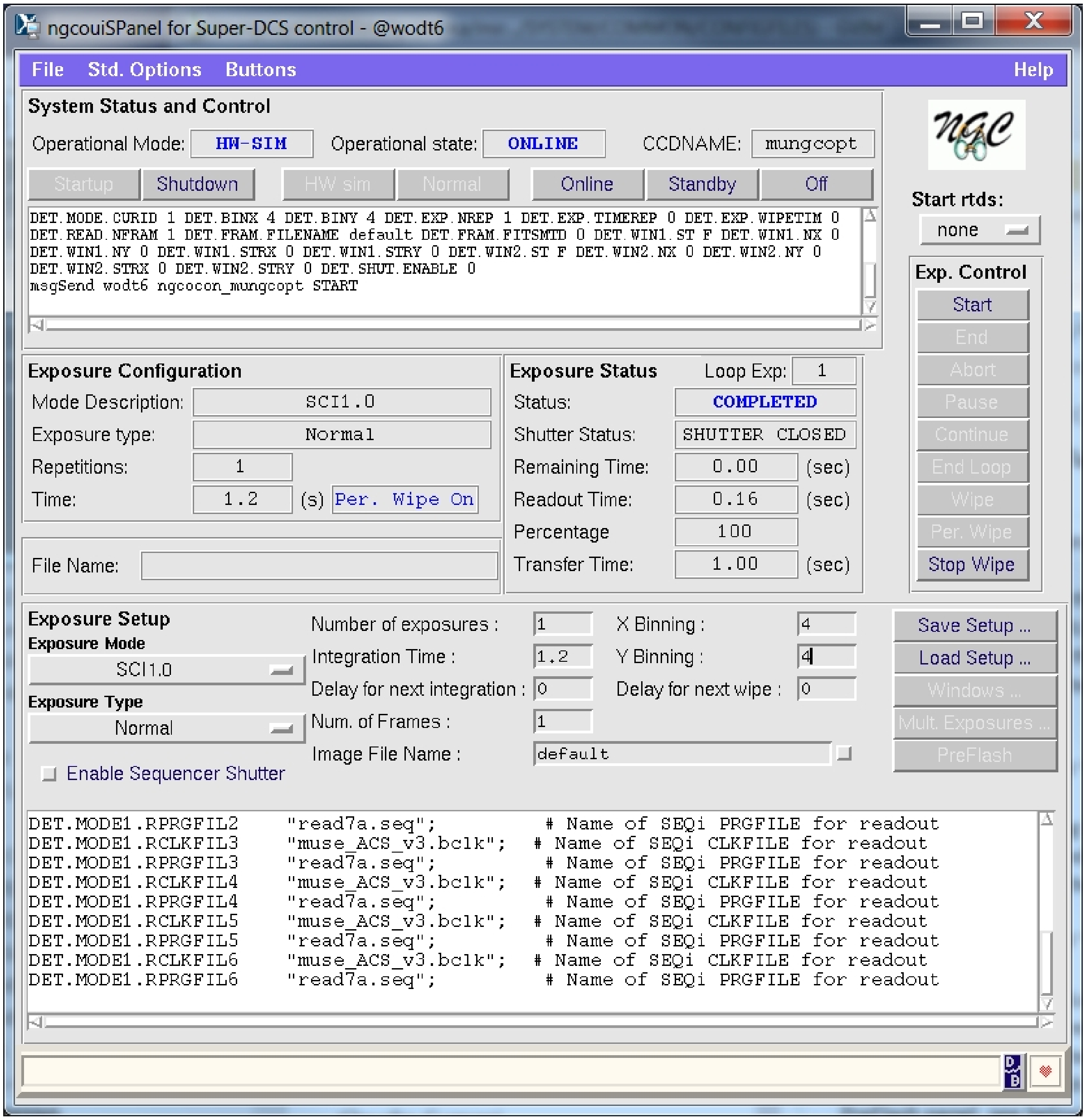
PFLASH2 : 0 (s)

Figure 6 - Subpanel for Preflash time setting

In the case of Multi-DCS systems (see Ch. 5), the Control Panel for Super Control Process must be used.

To startup the Panel run

```
ngcouiSPanel &
```



Operational Mode: HW-SIM Operational state: ONLINE CCDNAME: mungcopt

```
DET.MODE.CURID 1 DET.BINX 4 DET.BINY 4 DET.EXP.NREP 1 DET.EXP.TIMEREP 0 DET.EXP.WIPETIM 0
DET.READ.NFRAM 1 DET.FRAM.FILENAME default DET.FRAM.FITSMTD 0 DET.WIN1.ST F DET.WIN1.NX 0
DET.WIN1.NY 0 DET.WIN1.STRX 0 DET.WIN1.STRY 0 DET.WIN2.ST F DET.WIN2.NX 0 DET.WIN2.NY 0
DET.WIN2.STRX 0 DET.WIN2.STRY 0 DET.SHUT.ENABLE 0
msgSend wodt6 ngcocon_mungcopt START
```

Start rtds: none

Exposure Configuration

Mode Description: SCI1.0
Exposure type: Normal
Repetitions: 1
Time: 1.2 (s) Per. Wipe On
File Name:

Exposure Status

Loop Exp: 1
Status: COMPLETED
Shutter Status: SHUTTER CLOSED
Remaining Time: 0.00 (sec)
Readout Time: 0.16 (sec)
Percentage 100
Transfer Time: 1.00 (sec)

Exposure Setup

Exposure Mode: SCI1.0
Exposure Type: Normal
Number of exposures : 1
Integration Time : 1.2
Delay for next integration : 0
Num. of Frames : 1
Image File Name : default
X Binning : 4
Y Binning : 4
Delay for next wipe : 0

```
DET.MODE1.RPRGFIL2     "read7a.seq";          # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL3     "muse_ACS_v3.bclk";   # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL3     "read7a.seq";          # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL4     "muse_ACS_v3.bclk";   # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL4     "read7a.seq";          # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL5     "muse_ACS_v3.bclk";   # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL5     "read7a.seq";          # Name of SEQi PRGFILE for readout
DET.MODE1.RCLKFIL6     "muse_ACS_v3.bclk";   # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL6     "read7a.seq";          # Name of SEQi PRGFILE for readout
```

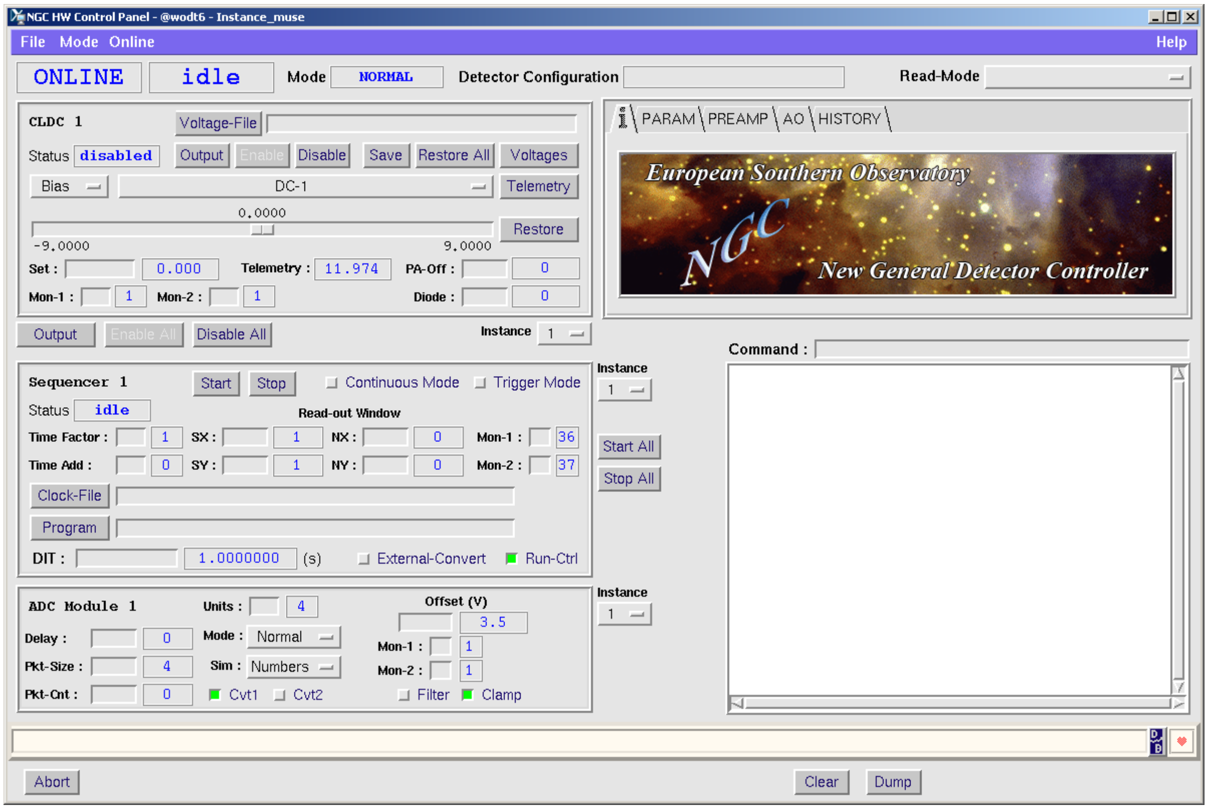
Figure 7 - Optical NGC Control Panel for the Super Control Process

14.2. Engineering Interface

A GUI panel is provided to help engineers in case of trouble (see Figure 8). It is evoked from the Control Panel and enables the most common operations needed for engineering.

The panel gives freedom to do actions at a low level and must be used with care !

It is assumed that the user knows the NGC sw and the VLT sw environment and is fully aware of the actions associated to each button and possible consequences.



NGC HW Control Panel - @wodt6 - Instance_muse

ONLINE idle Mode NORMAL Detector Configuration Read-Mode

CLDC 1
Status disabled
Bias DC-1
0.0000
-9.0000 9.0000
Set : 0.000 Telemetry : 11.974 PA-Off : 0
Mon-1 : 1 Mon-2 : 1 Diode : 0
Instance 1

Sequencer 1 Instance 1
Status idle Read-out Window
Time Factor : 1 SX : 1 NX : 0 Mon-1 : 36
Time Add : 0 SY : 1 NY : 0 Mon-2 : 37
DIT : 1.0000000 (s) □ External-Convert 🟩 Run-Ctrl

ADC Module 1 Units : 4 Offset (V) 3.5 Instance 1
Delay : 0 Mode : Normal Mon-1 : 1
Pkt-Size : 4 Sim : Numbers Mon-2 : 1
Pkt-Cnt : 0 🟩 Cvt1 □ Cvt2 □ Filter 🟩 Clamp

Command :

Figure 8 - NGC Engineer Panel

15. Special functionalities for Optical Instruments

15.1. Shutter Control by Pulpo Server

If shutter control is performed via the Pulpo Server (DET.SHUTi.CTRL=”pulpo” in the <label>CAMERA.cfg file, see 3.2.3.3), the device used to physically connect to the shutter is defined in the file pulpo.cfg, which must be archived in the instrument specific configuration module <xx>dcfg and must be correctly set.

Here is a self-explanatory example of a pulpo.cfg file for a system with 2 shutters connected via ttyc and ttyd:

```
#
# "@(#) $Id: pulpo.cfg,v 1.44 2004/05/10 22:47:31 vltsccm Exp $"
# --------------------------------------------------------------
#
# Pulpo configuration
#
# format is:  Pulpo_Unit_Number  Full_Device_Path
1 /dev/ttyc
2 /dev/ttyd
```

15.2. Temperature/pressure Monitoring

Unlike FIERA, NGCOPT does NOT provide facilities to monitor temperature and pressure values.

15.3. Adaptive Optics

NGCOPT does NOT provide facilities to operate Adaptive Optics.

16. Manpages

16.1.1. ngcoDcsOldb

```
NAME
       ngcoDcsOldb.sh  - Install and generate the online database environment.


SYNOPSIS
       ngcoDcsOldb -host <IWS | LLCU> -renv <renv> [<renv2> [<renv3> <...>]]]
                         [-instance <ccdname>] [-env <env>] [-user <user>]


DESCRIPTION
       On both the Instrument Workstation (IWS) and the NGC Linux LCU (LLCU)
       this shell script preliminary performs a system check (definition
       of environment variables, definition of local and remote
       environments on the local machine and in the ACC server, scanning,
       user running the software, etc.).
       On an IWS this shell script then installs in the directory
          $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl/
       the template files (DATABASE.db.NGCOSW and USER.db.NGCOSW)
       which can be used to generate the online database for
       an optical NGC system.
       On a LLCU this shell script generates and starts the online
       database environment.

       -host <IWS|LLCU>    Defines if the database must be generated
                           on an IWS or on A NGC LLCU

       -renv <renv>        name of remote online database environment
                           (on the IWS this is $CCDLENV, on the NGC LLCU
                           this is the $RTAPENV of the IWS).
                           ONLY IN THE "IWS" CASE, more <renv> can be given,
                           to check if they are all known by the
                           ACC server.

       -instance <ccdname> detector name (default $CCDNAME)
       -env <env>          name of local online database environment
                           (default $RTAPENV).

       -user <user>        name of the user running NGCOSW

FILES
       Source files:
         $VLTTOP/ENVIRONMENTS/ngco/DATABASE.db
         $VLTTOP/ENVIRONMENTS/ngco/USER.db
       Generated files on IWS:
         $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl/DATABASE.db.NGCOSW
         $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl/USER.dbNGCOSW
       Generated files on LLCU:
         $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl/DATABASE.db
         $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl/USER.db

ENVIRONMENT
       CCDNAME             CCD camera name
       RTAPENV             Online database environment name

RETURN VALUES
       0 if SUCCESS
       1 if FAILURE

EXAMPLES
       > ngcoDcsOldb -host LLCU -renv wte98
       > ngcoDcsOldb -host IWS -renv wodt8
       > ngcoDcsOldb -host IWS -renv wodt6 wodt8 -instance mycam -user myuser
```

16.1.2. ngcoDcsInstall

```
NAME
       ngcoDcsInstall.sh - Install NGCOSW files in INS_ROOT



SYNOPSIS
       ngcoDcsInstall -config <detector_module> [- root <ins_root>]
                      [-usr <ins_usr>] [-instance <ccdname>]
                      [-env <rtapenv>] [-lenv <ccdlenv>]
                      [-opmode <NORMAL | HW-SIM>] [-keep]

DESCRIPTION
       This shell script installs all files needed to run an
       optical NGC system (configuration files, CCD voltages, clock
       patterns and sequences) in the instrument directory
       <ins_root>/SYSTEM/COMMON/CONFIGFILES/

       -config <detector_module>   Name of the detector module containing
                                   the detector configuration.
                                   (xxopt is the optical detector template)

       -root <ins_root>            Root directory for the instrument the
                                   NGCOPT belongs to
                                   Default: $INS_ROOT (env. variable)

       -usr <ins_user>             User directory for the instrument the
                                   NGCOPT belongs to
                                   Default: $INS_USER (env. variable)

       -instance <ccdname>         detector name.
                                   Default: $CCDNAME (env. variable)

       -env <rtapenv>              Local online database environment
                                   to be used
                                   Default: $RTAPENV (env. variable)

       -lenv <ccdlenv>             Remote online database environment
```

```
                                   to be used
                                   Default: $CCDLENV (env. variable)

       -opmode <opmode>            NGCOSW operational mode
                                   Default: "NORMAL"

       -keep                       do not resolve environment variables
                                   defined in config files, if any.
                                   (Normally used only for tat tests)

FILES
       Source files:
       $VLTTOP/config/??dcfgCONFIG.cfg  ctoo configuration file for
                                        the optical NGC detector.
       $VLTTOP/config/??dcfgDCS.cfg     DCS configuration file for
                                        the optical NGC detector.
       $VLTTOP/config/??dcfgCAMERA.cfg  system configuration file for
                                        the optical NGC detector.
       $VLTTOP/config/??dcfgCAMERA_TARGET.cfg  target configuration file for
                                        the optical NGC detector.
       $VLTTOP/config/??dcfg/*          Files defining the detector
                                        voltages, clock patterns
                                        and sequences.

       Optical detector template files:
       $VLTTOP/templates/forNGC/xxopt/xxoptCONFIG.cfg
                                        ctoo configuration file for
                                        the optical NGC detector.
       $VLTTOP/templates/forNGC/xxopt/xxoptDCS.cfg
                                        DCS configuration file for
                                        the optical NGC detector.
       $VLTTOP/templates/forNGC/xxopt/xxoptCAMERA.cfg
                                        system configuration file for
                                        the optical NGC detector.
       $VLTTOP/templates/forNGC/xxopt/xxoptCAMERA_TARGET.cfg
                                        target configuration file for
                                        the optical NGC detector.
       $VLTTOP/templates/forNGC/xxopt/*
                                        Files defining the detector
```

```
                                 voltages, clock patterns
                                 and sequences.

       Destination files:
       <ins_root>/SYSTEM/COMMON/CONFIGFILES/${CCDNAME}CONFIG.cfg
       <ins_root>/SYSTEM/COMMON/CONFIGFILES/${CCDNAME}DCS.cfg
       <ins_root>/SYSTEM/COMMON/CONFIGFILES/${CCDNAME}CAMERA.cfg
       <ins_root>/SYSTEM/COMMON/CONFIGFILES/${CCDNAME}CAMERA_TARGET.cfg
       <ins_root>/SYSTEM/COMMON/CONFIGFILES/${CCDNAME}/*

ENVIRONMENT
       CCDNAME                    CCD camera name
       INS_ROOT                   default instrument root directory
       INS_USER                   default to SYSTEM

RETURN VALUES
       0 if SUCCESS
       1 if FAILURE

EXAMPLES
       > ngcoDcsInstall -config opdcfg
       Install all what needed for scientific CCD whose configuration
       files are stored in the "opdcfg" module
```

16.1.3. ngcoDcsStart

```
NAME
       ngcoDcsStart - startup of optical NGC DCS

SYNOPSIS
       ngcoDcsStart [-instance <ccdname>] [-env <env>] [-lenv <lenv>]
                    [-opmode <opmode>] [-gui <guiname>] [-xterm <T|F>]
                    [-autonlin <T|F>] [-verbose <level>] [-itc <itcproc>]
                    [-osc <oscproc>] [-kill]

DESCRIPTION
       This shell script performs a startup of NGC optical DCS.

       -instance <ccdname> detector name (default $CCDNAME)

       -env <env>          name of workstation online database environment
                           (default $RTAPENV).

       -lenv <lenv>        name of remote online database environment
                           (default $CCDLENV)
                           If lenv=0, only the NGC LCU processes are started.

       -opmode <opmode>    NGCOSW operational mode (default "NORMAL")
                           Valid values are:
                           NORMAL        - Normal Operational Mode (Default)
                                           NGC HW is used,
                                           NGC SW runs on Instrument
                                           Workstation (IWS) and NGC-LCU
                           HW-SIM        - HW is simulated

       -gui [<guiname>]    Launch the specified grafical user interface.
                           If <guiname>=NONE, no gui is started.
                           If no <guiname> is given, the default ngcouiPanel
                           is used.
                           At the moment, only the default program ngcouiPanel
                           is used, independently from the process name which
                           is specified.
```

```
-xterm <T|F>        When set to T, all processes are started
                    in new xterminals.
                    Default is "F".

-autonlin <T|F>     When set to T, the detector system automatically
                    goes to ONLINE at startup.
                    Default is "F".

-verbose <level>    Verbose level. Prints out system messages to the
                    standard output. The level gives the detail of the
                    messages. Default value is 0 (= no verbose output).

-itc <itcproc>      Name of the image transfer client process to be
                    launched in alternative to NGC Image Transfer Client.
                    Default is "" (ngcoitc is used).

-osc <oscproc>      Name of the super control process to be
                    launched in alternative to NGC Super Control Process.
                    Default is "" (ngcosc is used).

-sdcs               Start only the Super Control Process of a multi-DCS
                    instrument (By default NOT).
                    NOTE: This option is checked only when starting a
                    multy-dcs system.

-subsys             Start the system as a subsystem of a multi-DCS
                    instrument (By default NOT).
                    NOTE: This option shall NOT be explicitely used
                    (it is used only internally, when the utility calls
                    itself to start multi-DCS systems)

-hblocks <blocks>   Number of blocks in the header of the FITS files.
                    This option is useful when starting the subsystems
                    of a multi-DCS instrument, in order to configure them
                    all consistently.

-kill               kill all already running processes before starting
```

```
ENVIRONMENT
       CCDNAME        default for camera name (e.g. myccd)
       RTAPENV        default for WS local environment (e.g. myws)
       CCDLENV        default for LCU environment (e.g myngc)
       INS_ROOT       default root directory for instrument data

RETURN VALUES
       0 if SUCCESS
       1 if FAILURE

CAUTIONS
        .rhosts file on LCU system must contain user and hostname
       where this script runs, since it performs remote shell commands

EXAMPLES
       > ngcDcsStart -instance myccd -env myws -lenv myngc
           Start the NGCOSW for camera "myccd",
           WS environment "myws", LCU environment "myngc".

       > ngcDcsStart -instance myccd -env myws -lenv 0
           Start only the NGCOSW LCU processes
           for camera "myccd", WS environment "myws".

       > ngcDcsStart -instance myccd -env myws -lenv myngc -kill -gui
           Kill and restart the NGCOSW for camera "myccd",
           WS environment "myws", LCU environment "myngc".
           Gui is also started

SEE ALSO
       ngcoDcsStop, ngcoGetProcNum
```

16.1.4. ngcoDcsStop

```
NAME
       ngcoDcsStop - shut-down optical NGC DCS

SYNOPSIS
       ngcoDcsStop [-instance <ccdname>] [-env <env>] [-lenv <lenv>] [-kill]

DESCRIPTION
       This shell script performs a shut-down of NGC optical DCS.
       It does the following steps:
         1 - Verify if the main process is running.
         2 - Try to terminate NGC optical processes in a 'soft' way
             (command EXIT)
         3 - Try to terminate NGC optical processes in a 'hard' way
             (kill) - optional

       -instance <ccdname> detector name (default $CCDNAME)

       -env <env>          name of workstation online database environment
                           (default $RTAPENV).
                           If value is "FALSE", no action on WS part
                           of NGCOSW is taken

       -lenv <lenv>        name of remote online database environment
                           (default $CCDLENV)
                           If value is "FALSE", no action on LCU part
                           of NGCOSW is taken

       -kill               kill all processes

       -killpulpo          kill also pulpo server

       -sdcs               Stop only the Super Control Process of a multi-DCS
                           instrument (By default NOT).


ENVIRONMENT
```

```
       CCDNAME  default for camera name (e.g. myccd)
       RTAPENV  default for WS local environment (e.g. myws)
       CCDLENV  default for LCU environment (e.g myngc)

RETURN VALUES
       0 if SUCCESS
       1 if FAILURE

CAUTIONS
       a) .rhosts file on LCU system must contain user and hostname
          where this script runs, since it performs remote shell commands
       b) The "-kill" options should be used with care. By killing
          processes 'blindly', the system could remain in a dangerous
          state. To be used only to recover when the system gets stuck.

EXAMPLES
       > ngcDcsStop -instance myccd -env myws -lenv myngc
           Terminate in a 'soft' way the NGCOPT both at WS and LCU level
           for camera "myccd", WS environment "myws",
           LCU environment "myngc"

       > ngcDcsStop -instance myccd -env myws -lenv myngc -kill
           Terminate in a 'hard' way the NGCOPT both at WS and LCU level
           for camera "myccd", WS environment "myws",
           LCU environment "myngc"

       > ngcDcsStop -instance myccd -env myws -lenv FALSE -kill
           Terminate in a 'hard' way the NGCOPT at WS level only
           for camera "myccd", WS environment "myws"

       > ngcDcsStop -instance myccd -env FALSE -lenv myngc
           Terminate in a 'soft' way the NGCOPT at LCU level only
           for camera "myccd", LCU environment "myngc"


SEE ALSO
       ngcoDcsStart
```

16.1.5. ngcoGetProcNum

```
NAME
       ngcoGetProcNum - get process number

SYNOPSIS
       ngcoGetProcNum <envName> <procName>

DESCRIPTION
       Utility to retrieve the process number for the indicated
       process from the environment in which it is running.
       It is used by the script ngcoDcsStart.
       <envName>     environment name
       <procName>    process name

RETURN VALUES
       Process number if SUCCESS
       0 if FAILURE

EXAMPLES
       > ngcoGetProcNum $RTAPENV ngcoexp_$CCDNAME ; echo $?

SEE ALSO
       ngcoDcsStart
```

16.1.6. ngcoDcsClean

```
NAME
       ngcoDcsClean - Clean oldb envirionment and shared memory

SYNOPSIS
       ngcoDcsClean

DESCRIPTION
       This shell script performs a shutdown of the online database
       environment (defined by $RTAPENV), removing shared memory segments
       and zombie processes.

ENVIRONMENT
       RTAPENV        default for oldb environment

RETURN VALUES
       0 if SUCCESS
       1 if FAILURE

EXAMPLES
       > ngcoDcsClean
```

16.1.7. ngcoDcsTemplate

```
NAME
       ngcoDcsTemplate.sh - Install and generate template module.

SYNOPSIS
       ngcoDcsTemplate -template <template> -module <module>

DESCRIPTION
       Generate a new module from a template.
       To be used to generate a new image transfer process implementing
       image data post-processing or a super control process implementing
       new actions for standard commands.

       -template <template>  Template to start from
                             It must exist in $VLTTOP/templates/forNGC

       -module <module>      Module to be generated from template

FILES
       $VLTTOP/templates/forNGC/template

RETURN VALUES
       0 if SUCCESS
       1 if FAILURE

EXAMPLES
       Create module to implement image data post-processing
       > ngcoDcsTemplate -template xxxoitc -module newoitc
       Create new super control process
       > ngcoDcsTemplate -template xxxosc -module newosc
```

17. Example of NGCOPT usage

Assuming that we are using an XX system (*xxdcfg* instrument module) and that we want to pass relevant parameters using environment variables "à la FIERASW", in the following example the NGCOPT is started and some exposures are performed.

1. Start NGCOPT from the Instrument Workstation

```
ngcoDcsStart -instance $CCDNAME -env $RTAPENV -lenv $CCDLENV \
             -kill
```

2. Put NGCOPT in STANDBY

```
msgSend $RTAPENV ngcocon_$CCDNAME STANDBY ""
```

3. Put NGCOPT ONLINE

```
msgSend $RTAPENV ngcocon_$CCDNAME ONLINE ""
```

4. Perform periodic wiping

```
msgSend $RTAPENV ngcocon_$CCDNAME STARTWP ""
```

5. Prepare the next exposure (set exposure mode, type, time and binning)

```
msgSend $RTAPENV ngcocon_$CCDNAME SETUP \
  "-function DET.MODE.CURID 1 DET1.EXP.TYPE Normal \
  DET1.UIT1 10 DET1.BINX 1 DET1.BINY 1"
```

6. Start the exposure

```
msgSend $RTAPENV ngcocon_$CCDNAME START ""
```

7. Wait until the exposure has been completed

```
msgSend $RTAPENV ngcocon_$CCDNAME WAIT ""
```

8. Check the exposure status

```
dbRead "<alias>${CCDNAME}:exposure:control.state"
```

9. Prepare the next exposure (change exposure mode, type, time and binning)

```
msgSend $RTAPENV ngcocon_$CCDNAME SETUP \
  "-function DET1.MODE.CURID 3 DET1.EXP.TYPE Dark \
  DET1.UIT1 20 DET1.BINX 2 DET1.BINY 2"
```

10. Prepare the next exposure (define a loop of exposures)

```
msgSend $RTAPENV ngcocon_$CCDNAME SETUP \
  "-function DET1.EXP.NREP 10"
```

11. Start the loop of exposures

```
msgSend $RTAPENV ngcocon_$CCDNAME START ""
```

12. Wait until the last exposure has been completed

```
msgSend $RTAPENV ngcocon_$CCDNAME WAIT ""
```

13. Check the exposure status

```
dbRead "<alias>${CCDNAME}:exposure:control.state"
```

14. Prepare the next exposure (single exposure)

```
msgSend $RTAPENV ngcocon_$CCDNAME SETUP \
  "-function DET1.EXP.NREP 1"
```

15. Start the exposure

```
msgSend $RTAPENV ngcocon_$CCDNAME START ""
```

16. Pause the exposure

```
msgSend $RTAPENV ngcocon_$CCDNAME PAUSE ""
```

17. Modify the exposure time

```
msgSend $RTAPENV ngcocon_$CCDNAME SETUP \
  "-function DET1.UIT1 60"
```

18. Continue the exposure

```
msgSend $RTAPENV ngcocon_$CCDNAME CONT ""
```

19. Wait until the exposure has been completed

```
msgSend $RTAPENV ngcocon_$CCDNAME WAIT ""
```

20. Check the exposure status

```
dbRead "<alias>${CCDNAME}:exposure:control.state"
```

21. Stop periodic wiping

```
msgSend $RTAPENV ngcocon_$CCDNAME STOPWP ""
```

22. Exit

```
ngcoDcsStop -kill
```

Appendix A. Change LLCU's address and hostname

Whenever the IP address and the hostname of an LLCU have to be modified, the following steps must be performed:

A. On the system files (as "root")

- `/etc/hosts`

  Modify IP address, hostname and hostname aliases

- `/etc/sysconfig/network`

  Modify `HOSTNAME` and `GATEWAY`

- `/etc/sysconfig/network-scripts/ifcfg-eth<N>`

  (where <N> is 0 or 1, depending on which Ethernet port is used)

  Modify `IPADDR` (and `NETMASK`, if needed)

- `/etc/resolv.conf`

  Modify `search` and `nameserver`

B. On the pecs files

- Rename the file (as "pecsmgr")

  `/etc/pecs/releases/000/etc/locality/apps-<oldhostname>.env`

  as

  ``/etc/pecs/releases/000/etc/locality/apps-`hostname`.env``

  If needed, modify `RTAPENV`

- If needed, edit the file (as "pecsmgr")

  `/etc/pecs/releases/000/etc/locality/apps-all.env`

  and modify `ACC_HOST`

- Rename the file (as <ngcuser>, i.e., the user running NGCOPT)

  `~<ngcuser>/.pecs/apps-<oldhostname>.env`

  as

  ``~<ngcuser>/.pecs/apps-`hostname`.env``

- Rename the file (as <ngcuser>)

  `~<ngcuser>/.pecs/misc-<oldhostname>.env`

  as

  ``~<ngcuser>/.pecs/misc-`hostname`.env``

C. On the online database RTAPENV

- If `ACC_HOST` is local (`ACC_HOST=`hostname`` in `/etc/pecs/releases/000/etc/locality/apps-all.env`), reconfigure the ACC database as described in the “VLT Common Software – Installation Manual” (VLT-MAN-ESO-17200-0642), Chapter 4.3, starting from point 6. (modify the IP address, hostname and `RTAPENV` definitions in the plain ASCII data file `$VLTDATA/msql/accData.sql`)

- If `ACC_HOST` is not local, be sure that the ACC database on `$ACC_HOST` has been updated.

- If the `RTAPENV` name has been modified, then

  o Create the new `RTAPENV` directory

  ```
  vccEnvCreate –e $RTAPENV
  ```

  o Prepare the database

  ```
  cd $VLTDATA/ENVIRONMENTS/$RTAPENV/dbl/
  cp $VLTDATA/ENVIRONMENTS/<OLDRTAPENV>/dbl/*.db .
  make clean db
  ```

  o Init `$RTAPENV`

  ```
  vccEnvInit –e $RTAPENV
  ```

IMPORTANT: be sure that the setting of hostname and IP address are consistent in `/etc/host`, `/etc/sysconfig/network`, `/etc/sysconfig/network-scripts/ifcfg-eth0` and in the ACC database.

Then reboot the LLCU.

Appendix B. Troubleshooting

- **RTAPENV does not automatically start at startup**

  As “root” run

  ```
  /etc/init.d/rtap start
  ```

  If an error message like the following appears:

  ```
  Starting CCSLite environments: -bash:/tmp/<ENV>: Permission
  denied
  ```

  then run

  ```
  rm –f /tmp/<ENV>
  ```

- **Error “Permission denied” appears at startup and NGCOPT does not start**

  Check that `$HOME/.rhosts` is properly set (see 2.3.5)

- **At startup some remote process “does not reply to PING” (error message in the logMonitor)**

  Check that `$HOME/.rhosts` is properly set (see 2.3.5)

__oOo__