THE NEW GENERAL DETECTOR CONTROLLER - NGC/OPT

Claudio Cumani

Instrumentation SW Workshop 2008 - 2008, October 7

Differences btw IR and OPT detector controllers: intrinsic

- **“Exposure” handling**
  - **Optical**

    Rigid scheme for exposures (*wipe* - *integrate* - *[“move charges on detector while integrating”] - read)*.

    Active intervention of the control-server during the exposure is required (application of new voltages in each state).

    “Active” interface to different kinds of shutter controllers (open/close, status check, open/close delays, etc.).
  - **Infrared**

    Detector continuously read-out (infinite loop).

    Starting an exposure = starting transfer and storage of data. Once exposure is started, control server mainly reacts passively on incoming data-frames.

    No “active” interface to external devices (interfaces through trigger signals, e.g., for *nodding*).

Differences btw IR and OPT detector controllers: intrinsic

- **Data handling**
  - **Infrared**

    Computationally intensive different data pre-processing, read-out mode dependent.
  - **Optical**

    Detector read-out just once at the end of an exposure. The only processing to be done is pixel sorting and offset calibration (centroiding and bias-subtraction on request).

Differences btw IR and OPT detector controllers: historical

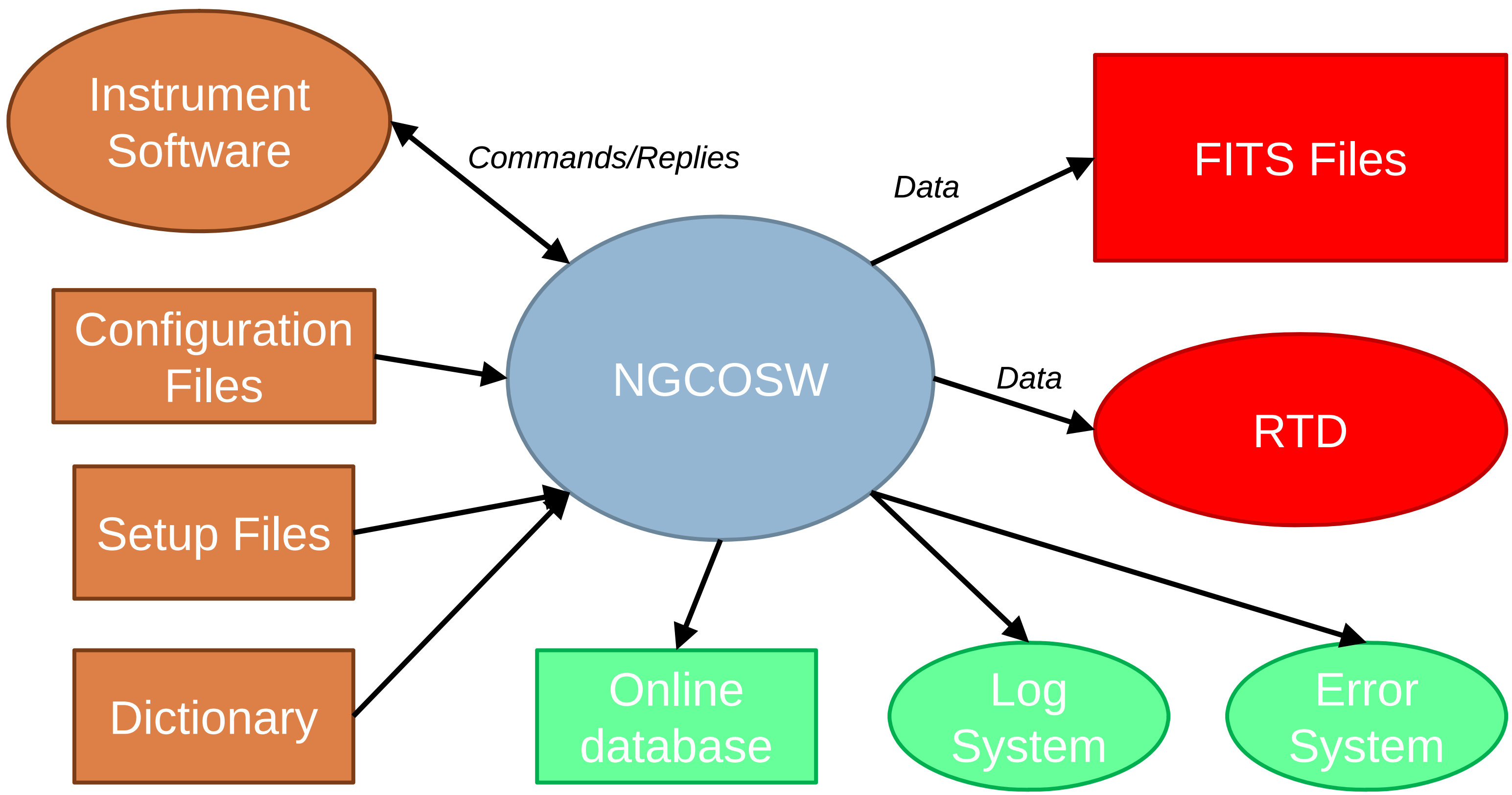
- optical detector controllers are requested to interface/control also devices which are not – strictly speaking - part of the detector, like vacuum and temperature control (and write values in FITS file header)

Optical NGC needs its own software

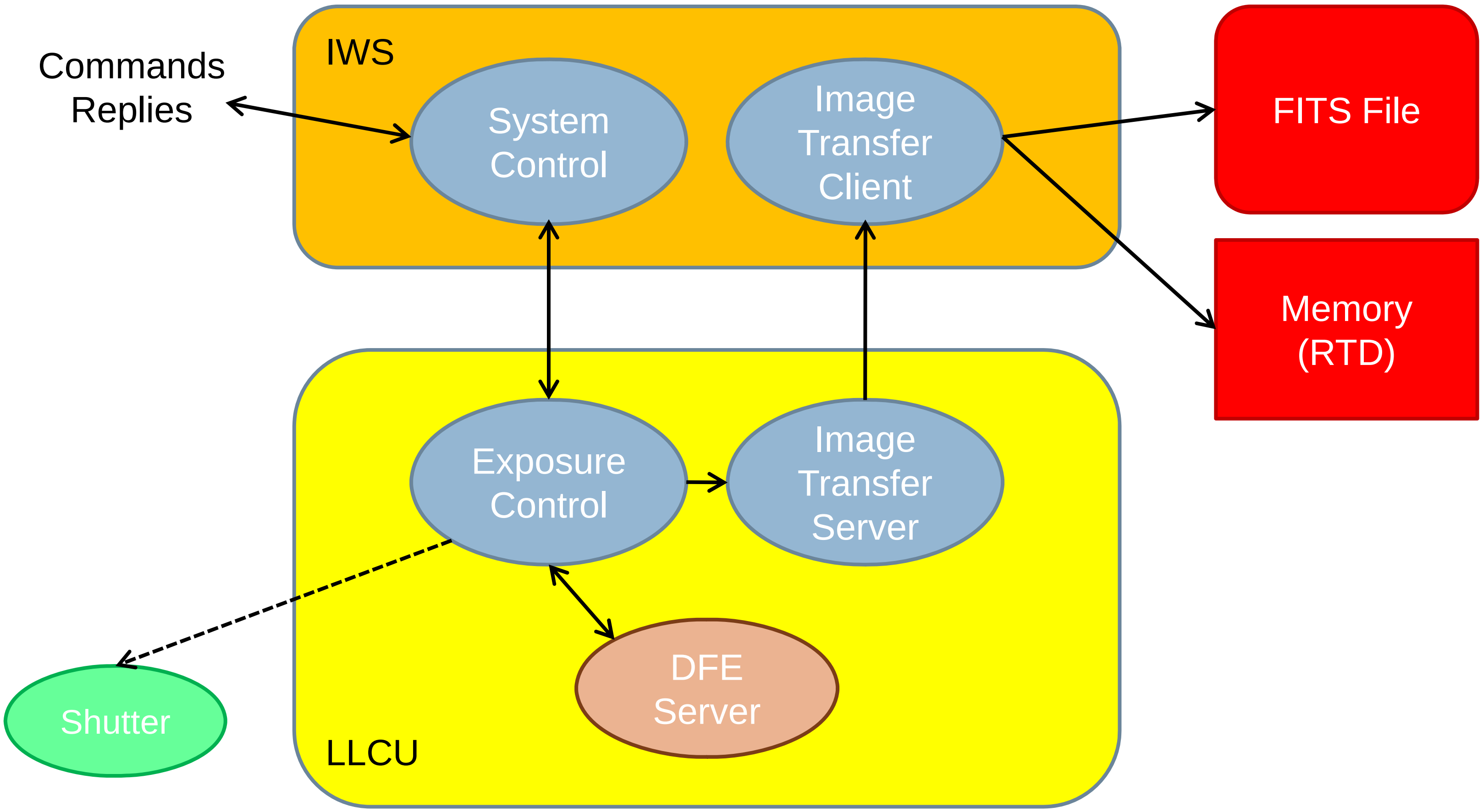
- Base software common to Infrared and Optical detectors to interface the hardware (thanks Joerg)
- At higher level:

  NGCIRSW and **NGCOSW**

NGC Software Environment



NGCOSW Processes



Operational Modes

- **Normal mode**: the NGC detector electronics is connected.
- **Hardware-Simulation mode**: the NGC detector electronics is simulated.
  - The FIERA LCU-Simulation mode is equivalent to Hardware-Simulation.
  - The NGCOSW can either be distributed on both the IWS and the NGC LLCU or run completely on one of the two platforms.

Optical “Exposures Modes”

- **Exposure Modes** define the set of voltages, clock patterns and sequences to be applied to the different “steps” of an exposure (wipe, integrate, read).
- Same approach of FIERA
- Exposure Modes are defined in the detector Configuration File (different for each instrument)

Temperature/vacuum monitoring

3 options under discussion:

- Via NGC LLCU serial port (à la FIERA)
- Via standard LCU (à la IRACE)
- Via Serial-to-ethernet adapter

NGCOSW code

- NGCOSW code generated using the wsf (workstation software framework) tool developed by SDD
- See: Andolfato L., Karban R., "*Workstation Software Framework*", article for SPIE 2008 "*Astronomical Telescopes and Instrumentation*" Conference, Marseille, Jun 23-28, 2008

NGCOSW overview

- ***ngcocon*** - The NGC System Coordination module for optical applications. This includes all required scripts for system startup and shutdown.
- ***ncgoctr*** - The NGC Exposure Control module for optical applications.
- ***ncgoexp*** - The NGC Exposure Coordination module for optical applications.
- ***ncgoits*** - The NGC Image Transfer Server module for optical applications.
- ***ncgoitc*** - The NGC Image Transfer Client module for optical applications.
- ***ngcoui*** - Engineering GUI used for direct system interaction and data acquisition.

The modules will be part of the **VLTSW Releases**.

All modules contain **Test Procedures** for **TAT** (automated testing).

Installation

NGCOSW is built on top of the NGCIRSW

- Via installation scripts
  - Install NGCIRSW first (cmm module ngcarch)

    cd <YOUR_SRC_DIR>; cmmCopy ngcarch; cd ngcarch/src/; make all install
  - Then install NGCOSW (cmm module ngcoarc)

    cd <YOUR_SRC_DIR>; cmmCopy ngcoarc; cd ngcoarc/src/; make all install
- ngcins software module contains a pkgin installation-configuration (for both NGC IR and OPT software):

  cd <YOUR_SRC_DIR>; cmmCopy ngcins; pkginBuild ngcins

Configuration - environment

- # Environment variables

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

- 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

Configuration - system check

ngcoDcsOldb preliminary check:

- Are the environment variables defined?
- Is ACC server defined and running?
- Are local and remote environments defined on the local computer and in the ACC server?
- Is scanning properly configured?
- Is the user which shall run the software defined on the local and the remote computer?
- …

Configuration - oldb

Create the environment: ngcoDcsOldb (different on IWS and LLCU)

- On the IWS: ngcoDcsOldb -renv $CCDLENV -host IWS

  NOTE: only template files `DATABASE.db.NGCOSW` and `USER.db.NGCOSW` are generated: use them to edit `DATABASE.db` and `USER.db`

  - Example 1: Instrument controlling one camera, add in `DATABASE.db`:

```
#undef  CCDNAME
#undef  ngcdcsINSTANCE
#undef  NGCROOT
#define CCDNAME <myCCDNAME>
#define ngcdcsINSTANCE ngcdcs_<myCCDNAME>
#define NGCROOT :Appl_data<:myPATH>:CCDNAME
```

Configuration - oldb

- Example 2: Instrument controlling four cameras, add in `DATABASE.db`:

```
#undef  DCSNAME
#undef  CCDNAME
#undef  ngcdcsINSTANCE
#undef  CCDNAME2
#undef  ngcdcsINSTANCE2
#undef  CCDNAME3
#undef  ngcdcsINSTANCE3
#undef  CCDNAME4
#undef  ngcdcsINSTANCE4
#undef  NGCROOT
#define DCSNAME <INSTRUMENT>
#define CCDNAME <myCCDNAME>
#define ngcdcsINSTANCE ngcdcs_<myCCDNAME>
#define CCDNAME2 <myCCDNAME2>
#define ngcdcsINSTANCE2 ngcdcs_<myCCDNAME2>
#define CCDNAME3 <myCCDNAME3>
#define ngcdcsINSTANCE3 ngcdcs_<myCCDNAME3>
#define CCDNAME4 <myCCDNAME4>
#define ngcdcsINSTANCE4 ngcdcs_<myCCDNAME4>
#define NGCROOT :Appl_data<:myPATH>:DCSNAME
```

Configuration - oldb

- Once the `DATABASE.db` and `USER.db` files have been properly edited on the IWS, 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

- On the LLCU: ngcoDcsOldb -renv <IWS_RTAPENV> -host LLCU

Configuration - oldb

- The basic structure of the database (on the IWS of a system with just one camera and on the LLCU) is as follows:

```
--o <alias><CCDNAME>--      |--o exposure (exposure state)
                            |--o ngcdcs   (device driver)
                            |--o ngcocon  (coordination process)
                            |--o ngcoctr  (control process)
                            |--o ngcoexp  (exposure handler)
                            |--o ngcoitc  (image transfer client)
                            |--o ngcoits  (image transfer server)
                            |--o system   (NGC system state)
                            |--o wcs      (world coordinate system)
```

Configuration - oldb

- On the IWS, the structure of the database of a system with more cameras (e.g., 4) is as follows:

```
--o <alias><DCSDNAME>--     |--o <camera1>     (first camera)
                            |--o <camera2>     (second camera)
                            |--o <camera3>     (third camera)
                            |--o <camera4>     (fourth camera)
                            |--o exposure (exposure state)
                            |--o ngcocon  (coordination process)
                            |--o ngcoitc  (image transfer client)
                            |--o system   (NGC system state)
                            |--o wcs      (world coordinate system)
```

where <camera1>, …, <camera4> have the basic structure shown before

Configuration – Instrument module

- Install the instrument module (<xxdcfg>)

  cmmCopy <xxdcfg>

  cd <xxdcfg>/src; make all install

The instrument module <xxdcfg> contains

- The voltages, patterns and sequences to drive the detector
- The detector startup configuration file **<xx>dcfgCONFIG.cfg** and the configuration set **<xx>dcfgCAMERA.cfg**

Note: naming convention: <xx> instrument specific, d=detector, cfg=configuration

Configuration – Instrument module

- Configuration set keywords specific to optical systems

```
######################################################################
# 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
```

Configuration – Instrument module

- Configuration set keywords specific to optical systems (cont.)

```
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
```

Configuration – Instrument module

- Configuration set keywords specific to optical systems (cont.)

```
####################################################################
# MODE description
####################################################################

DET.MODE1.NAME          "Test1";          # Exposure mode name
DET.MODE1.DESC          "Test mode 1";    # Exposure mode description
DET.MODE1.TRIGGER       F;                # Enable trigger
DET.MODE1.GAIN          "";               # Gain used
DET.MODE1.BNDWTH        "";               # Bandwidth used
DET.MODE1.WREP          1;                # Wipe sequence repetition number
DET.MODE1.WCLDFIL1      "wipe1.v";        # Name of CLDCi FILE for wipe
DET.MODE1.WCLKFIL1      "wipe1.bclk";     # Name of SEQi CLKFILE for wipe
DET.MODE1.WPRGFIL1      "wipe1.seq";      # Name of SEQi PRGFILE for wipe
DET.MODE1.PREP          1;                # Preint sequence repetition number
DET.MODE1.PCLDFIL1      "preint1.v";      # Name of CLDCi FILE for preintegration
DET.MODE1.PCLKFIL1      "preint1.bclk";   # Name of SEQi CLKFILE for preintegration
DET.MODE1.PPRGFIL1      "preint1.seq";    # Name of SEQi PRGFILE for preintegration
DET.MODE1.DREP          0;                # During int sequence repetition number
DET.MODE1.DCLDFIL1      "";               # Name of CLDCi FILE during integration
DET.MODE1.DCLKFIL1      "";               # Name of SEQi CLKFILE during integration
DET.MODE1.DPRGFIL1      "";               # Name of SEQi PRGFILE during integration
DET.MODE1.RREP          1;                # Readout sequence repetition number
DET.MODE1.RCLDFIL1      "read1.v";        # Name of CLDCi FILE for readout
DET.MODE1.RCLKFIL1      "read1.bclk";     # Name of SEQi CLKFILE for readout
DET.MODE1.RPRGFIL1      "read1.seq";      # Name of SEQi PRGFILE for readout
DET.MODE1.ADCSAMPL      "-1,1";           # ADC data sampling factors
```

Configuration – Instrument module

- Configuration set keywords specific to optical systems (cont.)

```
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    1.2;              # Readout noise per output (e-)

####################################################################
# SHUTTER description
####################################################################

DET.SHUT1.AVAIL        F;             # Shutter available or not
DET.SHUT1.CTRL         "ngc";         # Shutter controller
DET.SHUT1.TYPE         "iris";        # Shutter type
DET.SHUT1.ID           "eso-01";      # Shutter unique identifier
DET.SHUT1.DEVIDX       1;             # Device index
DET.SHUT1.ROUTE        "2";           # Route to module
DET.SHUT1.NAME         "Shutter-1";   # Optional module name
```

Configuration – INS_ROOT

- Populate the INS_ROOT with the files from the instrument module <xxdcfg>:

  ngcoDcsInstall -config <xxdcfg>

Data format

- Images are saved in $INS_ROOT/$INS_USER/DETDATA
- Images are saved as FITS files:
  - Using the “image extension per chip" format: data are ordered by detector, each detector corresponds to an extension. A primary header sits on the top of the file.
  - However, if the camera has only one detector, no extension is used.

The Bible: Data Interface Control Document, GEN-SPE-ESO-19400-0794 (last issue: 4.0, 08.04.2008)

Operation - Example

- Start NGCOSW from the Instrument Workstation and put it ONLINE

```
ngcoDcsStart -instance $CCDNAME -env $RTAPENV -lenv $CCDLENV –kill
msgSend $RTAPENV ngcocon_$CCDNAME STANDBY ""
msgSend $RTAPENV ngcocon_$CCDNAME ONLINE ""
```

- Prepare the exposure (set exposure mode, type, time and binning) and start it

```
msgSend $RTAPENV ngcocon_$CCDNAME SETUP \
    "-function DET.MODE.CURID 1 DET1.EXP.TYPE Normal \
    DET1.WIN1.UIT1 10 DET1.WIN1.BINX 1 DET1.WIN1.BINY 1“
msgSend $RTAPENV ngcocon_$CCDNAME START ""
```

- Wait until the exposure has been completed and then check status

```
msgSend $RTAPENV ngcocon_$CCDNAME WAIT ""
dbRead "<alias>${CCDNAME}:exposure:control.state"
```

Operation - Example (cont.)

- Prepare and start loop of 10 biases

```
msgSend $RTAPENV ngcocon_$CCDNAME SETUP \
    "-function DET1.EXP.NREP 10 DET1.EXP.TYPE Dark DET1.WIN1.UIT1 0 "
msgSend $RTAPENV ngcocon_$CCDNAME START ""
```

- Wait until the exposure has been completed and then check status

```
msgSend $RTAPENV ngcocon_$CCDNAME WAIT ""
dbRead "<alias>${CCDNAME}:exposure:control.state"
```

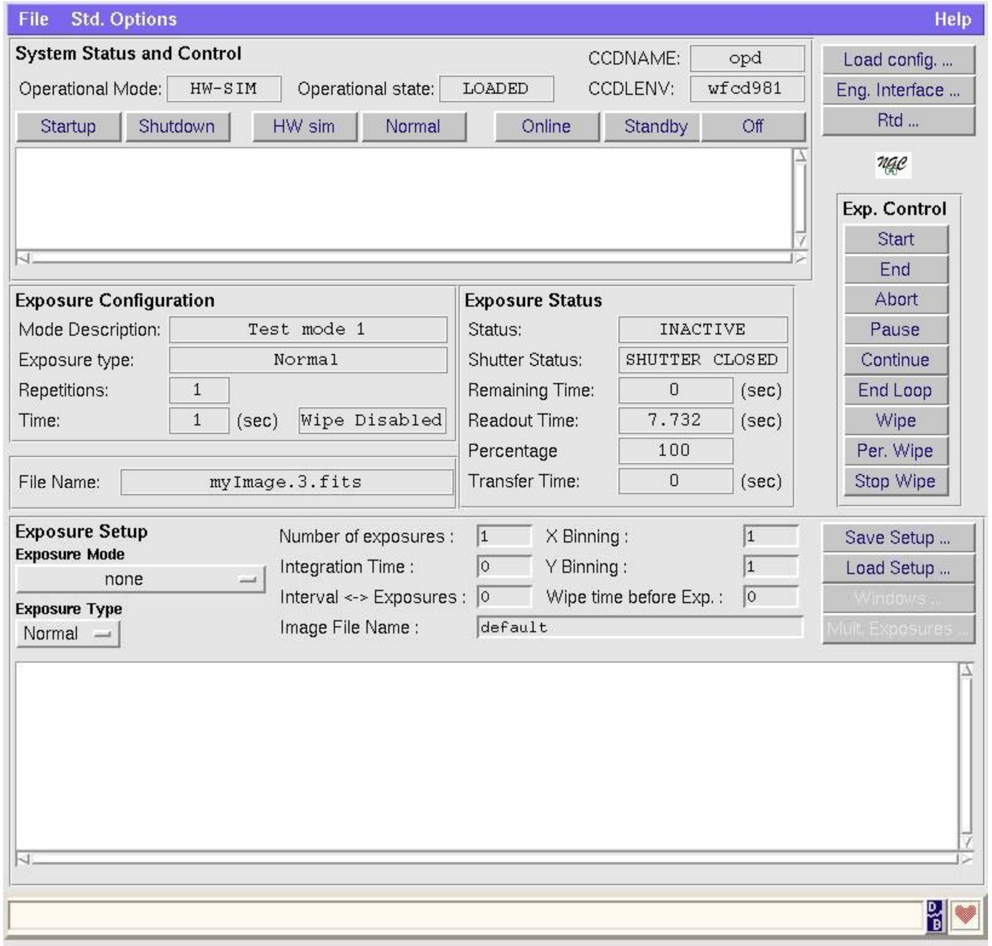
- Exit

```
ngcoDcsStop -kill
```

Multiple Instances of DCS

- The coordination control process ngcocon is the only command interface between ICS and NGCOSW.
- Even if multiple instances of DCS are used (e.g., for instruments which control more than one NGC-LLCU), the coordination control process is the only command interface between ICS and NGCOSW.

NGCOSW Graphical User Interface



Documentation

- VLT-MAN-ESO-13660-4510 NGC - User Manual
- VLT-MAN-ESO-13660-4085 NGC Infrared DCS - User Manual
- VLT-MAN-ESO-13660-4086 NGC Optical DCS - User Manual
- VLT-MAN-ESO-13660-4560 NGC-LCU Interface SW – User Manual
- VLT-LIS-ESO-13660-3907 NGC Project Glossary
- VLT-LIS-ESO-13660-3908 NGC Project Acronyms

Feedback

For feedback, questions, problem reporting: write to

ngc@eso.org