

KOSMOS ICD 3.1 Instrument Controller to Instrument System

P. N. Daly

National Optical Astronomy Observatories, 950 N. Cherry Avenue, P. O. Box 26732,
Tucson AZ 85726-6732, USA
pnd@noao.edu

Abstract. This document describes the interface between the instrument controller and the instrument system. Primarily, this identifies the various levels of interaction between the (GWC-based) instrument components controller and NOCS software.

Contents

1. Document Revision History	2
2. Purpose And Scope Of Document	2
3. WBS Element(s) and NOAO Account Code(s)	2
4. System Architecture	2
4.1. Message Sequence Chart	4
4.2. Hardware Configuration	4
5. Interface Description	4
5.1. Configuration Files	5
Filter Configuration File	5
Disperser Configuration File	7
Slit Configuration File	7
5.2. DRAMA Interface	7
configure	7
disperser	7
filter	7
camfocus	8
colfocus	8
help	8
init	8
slit	8
test	8
5.3. GWC Interface	8
5.4. Example(s)	8
testnics	8
A Instrument Control System GWC Interface	10

1. Document Revision History

6 July 2010: original version, (pnd@noao.edu).

9 December 2010: added section on configuration files, (pnd@noao.edu).

2. Purpose And Scope Of Document

This document identifies the interface between the instrument components controller and the instrument control system. It is intended for software and/or hardware engineers on either side of this divide to clearly identify their rôle within the overall system architecture and to implement any sub-system to achieve the desired objective.

This document is under (manual) revision control. All corrections, additions or comments should be addressed to the principal author.

3. WBS Element(s) and NOAO Account Code(s)

KOSMOS WBS v1.9^[1] shows this interface to be described by element D.N.9.5 (Spectrograph Interface) described therein as “design and coding of software to interface to the spectrograph control electronics and software”. The appropriate NOAO account code is N-MR210-D95.

4. System Architecture

The original implementation of the NOCS¹ is described elsewhere^[2]. The KOSMOS implementation of this architecture is shown in the cartoon of Figure 1 on page 3. The main elements of the NOCS are:

NGUI defines the observation scripting engine and is the *only* interface into the system. The motivating document behind NGUI is the *Operating Concepts Definition Document*^[3].

NICS (NOCS Instrument Control System) defines the interface to the instrument, specifically the hardware elements. The motivating documents behind NICS are ICD 3.1^[7] and ICD 3.2^[8].

NICC (NOCS Instrument Components Controller) defines the instrument-specific interface to the appropriate hardware. This entity is expected to be different for each instance of the NOCS (*e.g.*, NEWFIRM, Mosaic 1.1 and KOSMOS/COSMOS).

NSML (NOCS Monsoon Supervisor Layer) defines the interface to the detector system. The motivating document behind NSML is ICD 4.1 and describes the detailed interface of an implementation of the generic pixel server dictionary^[4].

NTCS (NOCS Telescope Control System) defines the interface to the telescope control system. The motivating document behind NTCS is the ICD 5.1^[6] (this document).

NOHS (NOCS Observation Header System) defines the interface to the environmental and observational meta-data which is published to the DHS. The motivating document behind NOHS is the ICD 6.1^[5].

The guiding principles behind the NOCS design are (were):

¹This section is common to several ICDs so that each ICD can be read as a standalone document.

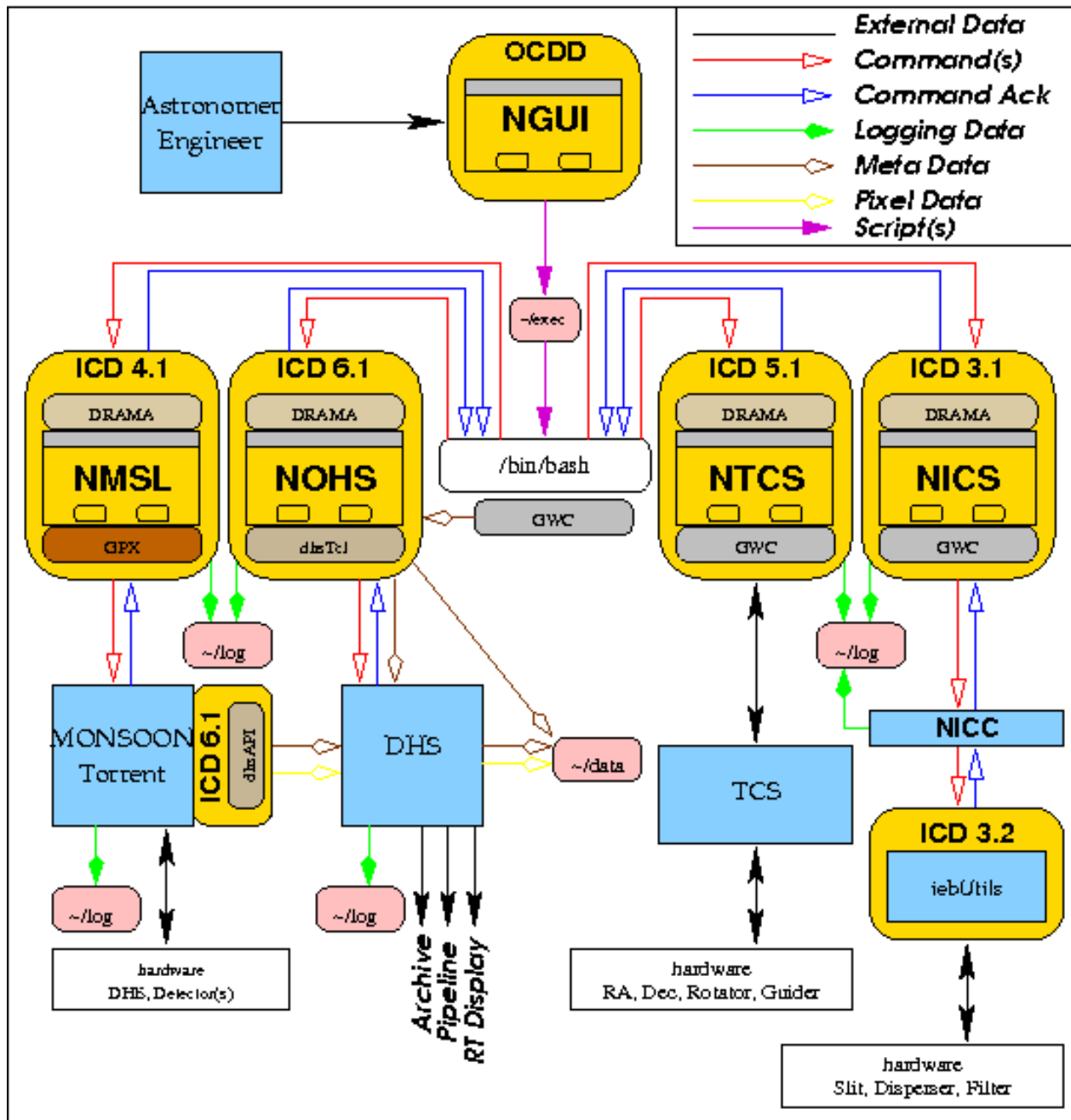


Figure 1.: KOSMOS Implementation of the NOCS Architecture by ICD

- *Simplicity*. The amount of coding required is the minimum necessary to achieve the science objective;
- *Modularity*. Only those parts of the system required at any specific instance need be started;
- *Supportability*. The system is written primarily in Tcl/tk which is widely used at the Mayall 4m;
- *Separability*. The system has clearly defined interfaces;
- *Repeatability*. Any script should be re-usable;
- *Consistency*. The science scripts ensure data is taken in a consistent manner with the minimum of human intervention and present ‘well known’ data products form ingestion downstream of the acquisition (*e.g.*, data analysis pipelines and/or data archives).
- *Logging*. Each task within the system should provide enough logging information for informal post-mortem debugging.

Further, various levels of simulation exist within the system so that laboratory testing can proceed without external hardware (or software) being present.

4.1. Message Sequence Chart

The fundamental unit of the *NOAO Observation Control System* is, by definition, the observation. This unit is mapped to the message sequence chart of Figure 2 on page 5. In short, it is the job of NGUI to re-produce a science-oriented recipe as an executable script and, essentially, create a series of linked observations to achieve the given science objective. Although the GUIs are written in Tcl/tk, the underlying re-scheduling mechanism is DRAMA^[9]. Thus, a script is a series of *ditscmd* directives to the other tasks in the system. The GUIs do the work ‘behind the scenes’ and return a simple SUCCESS or FAILURE status on completion of the action. If successful, the script continues. If not, the script aborts with an indication as to where in the script is failed and the task executing at that moment.

It should be heuristically obvious from the above, that the NOCS employs a demarcation paradigm inasmuch as NGUI will create a script but will *not* run it. Executing a script is a manual operation and clearly separates the creation of the script from its execution. A favourable side-effect is that NGUI is *not* required at the telescope and can be delivered to the astronomer prior to his/her observing run for self-paced familiarisation.

4.2. Hardware Configuration

Figure 3 on page 6 shows a typical hardware and network configuration of a NOCS implementation. Specifically, the PAN talks to the DHE (and hence the focal plane array) over an SL100 fiber interface but transfers the data to the primary DHS machine via the private network. Inside the same rack will be a KVM switch (TK-802R) and an Ethernet power controller (ECR2 DLI).

5. Interface Description

The KOSMOS instrument components controller to instrument control system interface has enough functionality to bring up the system in a known, safe, state ready to take data. The following commands are currently supported and there are no additions envisaged.

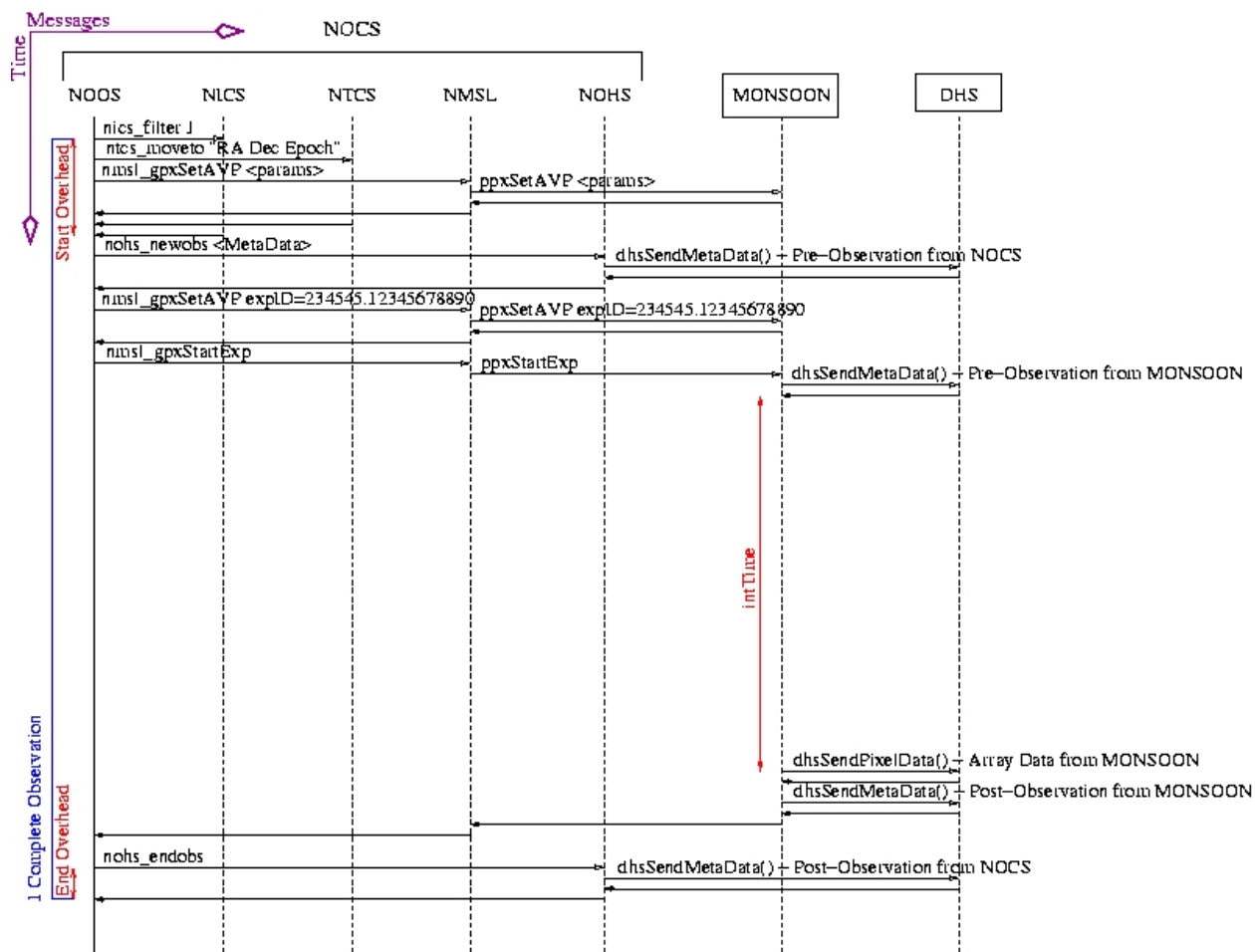


Figure 2.: NOCS Message Sequence Chart

5.1. Configuration Files

Both the NICS and NICC components of the software will read the same set of configuration files which may be edited, by observatory staff, during a semester. Therefore, the ability to re-read these configuration files on-the-fly without re-starting the software is vital.

Filter Configuration File An example filter configuration file is shown below. Note that these filters do exist at KPNO and have the correct serial numbers but may not necessarily be used in this way!

```
#+
# KOSMOS Filter Wheel List dd-mmm-yyyy for semester yyyyY
#
#NICC \NGUI \Combination \FW1 Pos\FW2 Pos\Serial Number(s)
#-
U \U \U + open \1 \6 \k1001+open
B \B \B Harris + open \2 \6 \k1002+open
V \V \V Harris + open \3 \6 \k1003+open
R \R \R Harris + open \4 \6 \k1004+open
I \I \I Nearly-Mould + open \5 \6 \k1005+open
g \g \open + g SDSS \6 \1 \open+k1017
```

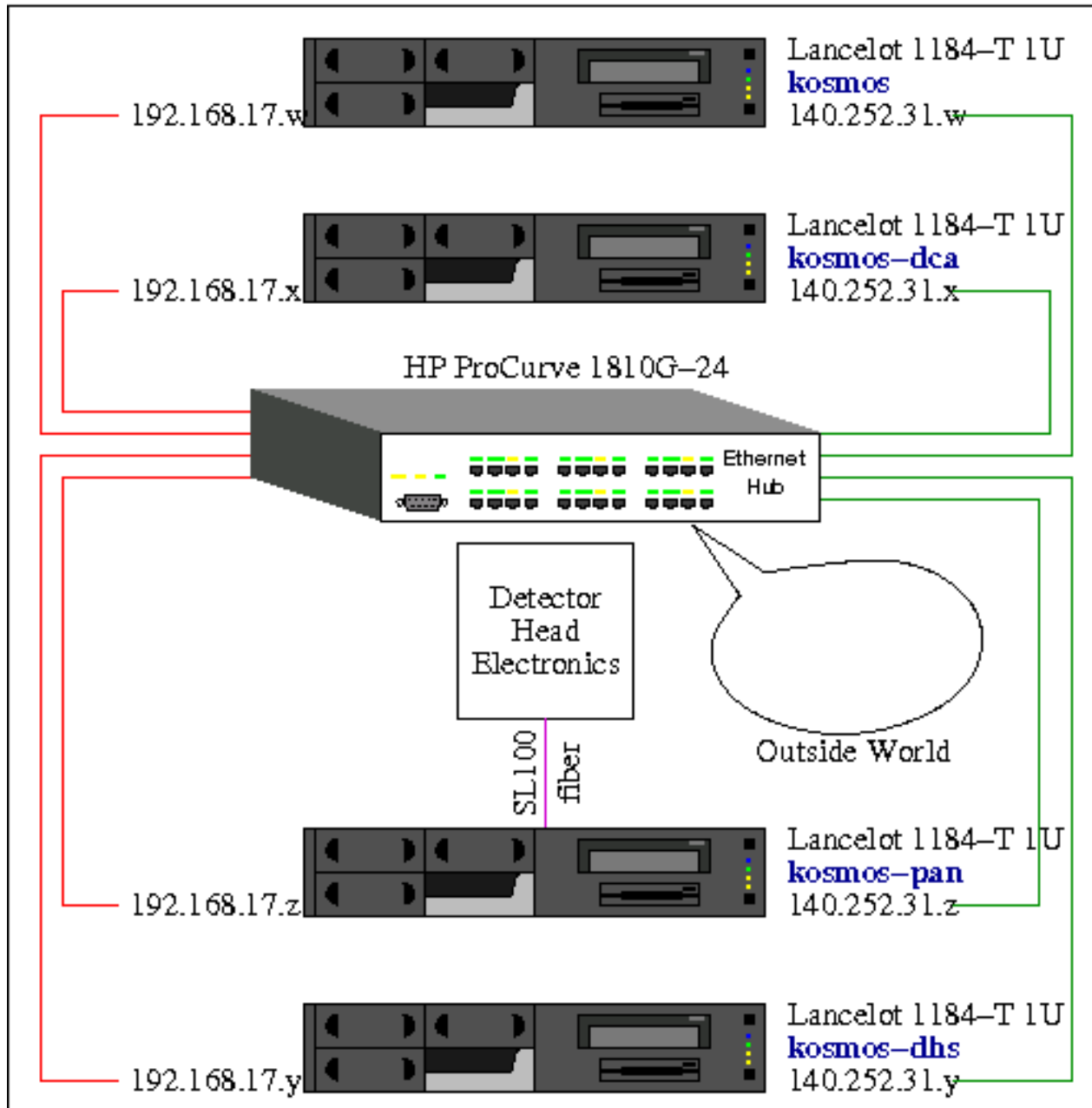


Figure 3.: Typical NOCS Hardware and Network Implementation

```

r      \r      \open      + r SDSS \6      \2      \open+k1018
i      \i      \open      + i SDSS \6      \3      \open+k1019
z      \r      \open      + z SDSS \6      \4      \open+k1020
colmask\colmask \open      + colmask \6      \5      \open+kmask
Open   \Open   \open      + open   \6      \6      \open+open

```

Disperser Configuration File An example disperser configuration file is shown below. Note that these dispersers do *not* exist so have dummy names and serial numbers!

```

#+
# KOSMOS Disperser Wheel List dd-mmm-yyyy for semester yyyyY
#
#NICC \NGUI \Comment \DW Pos\Serial Number(s)
#-
r2000 \r2000 \Red, R=2000, 800nm \1 \kr2000
b2000 \b2000 \Blue, R=2000, 500nm \2 \kb2000
g1300 \g1300 \Green, R=1300, 600nm \3 \kg1300
lowres \lowres \Low resolution, 200-900nm \4 \klowres
r3000 \r3000 \R=3000 \5 \kr3000
Open \Open \open \6 \open

```

Slit Configuration File An example slit configuration file is shown below. Note that these slits do *not* exist so have dummy names and serial numbers!

```

#+
# KOSMOS Slit Wheel List dd-mmm-yyyy for semester yyyyY
#
#NICC \NGUI \Comment \SW Pos\Serial Number(s)
#-
1pix \1pix \1 pixel 0.60 asec \1 \k1pix
2pix \2pix \2 pixel 0.90 asec \2 \k2pix
3pix \3pix \3 pixel 1.25 asec \3 \k3pix
4pix \4pix \4 pixel 2.50 asec \4 \k4pix
5pix \5pix \5 pixel 5.00 asec \5 \k5pix
Open \Open \open \6 \open

```

5.2. DRAMA Interface

configure This command sends a configure request to the instrument:

```
ditscmd nics nics_configure Argument1="name1 = val1,name2 = val2,name3 = val3"
```

where *name1/name2/name3* is an astronomer-friendly filter/disperser/slit name in any order.

Not all items need be present so:

```
ditscmd nics nics_configure Argument1="filter=name"
```

is equivalent to:

```
ditscmd nics nics_filter Argument1="name"
```

disperser This command sends a disperser demand to the instrument:

```
ditscmd nics nics_disperser Argument1="name"
```

where *name* is an astronomer-friendly disperser name.

filter This command sends a filter demand to the instrument:

```
ditscmd nics nics_filter Argument1="name"
```

where *name* is an astronomer-friendly filter name.

camfocus This command sends a focus demand to the instrument camera:

```
ditscmd nics nics_camfocus Argument1="value"
```

where *value* is expressed in μm .

colfocus This command sends a focus demand to the instrument collimator:

```
ditscmd nics nics_colfocus Argument1="value"
```

where *value* is expressed in μm .

help This command returns simple help text on the functionality contained within the interface:

```
ditscmd nics nics_help
```

init This command re-initializes the interface and returns the system interface to a ready state:

```
ditscmd nics nics_init
```

slit This command sends a slit demand to the instrument:

```
ditscmd nics nics_slit Argument1="name"
```

where *name* is an astronomer-friendly slit name.

test This command performs both the *init* and *help* actions to test the underlying communications path:

```
ditscmd nics nics_test
```

5.3. GWC Interface

The NICS is a GWC-aware^[10] client and subscribes to command and telemetry streams to enable the above instrument functionality. The instrument GWC interface is defined in appendix A.

5.4. Example(s)

testnics This is a wrapper that allows the end-user to test the system without creating a specific script. The simplest invocation is:

```
% testnics
```

which will give a list of instrument-specific options. Some examples are:

```
% testnics configure "filter=U,disperser=lo,slit=long"
```

```
% testnics disperser lo
```

```
% testnics filter U
```

```
% testnics camfocus 450.0
```

```
% testnics colfocus 11000.0
```

```
% testnics slit long
```

Acknowledgments. PND thanks Jay Elias, Mark Trueblood, Bob marshall and Shelby Gott for comments and corrections.

References

1. Elias, J., 2010, 'KOSMOS Work Breakdown Structure Summary', v1.11, Kosmos Project, NOAO.
2. Daly, P. N. et al., 2008, 'The NEWFIRM Observing Software: From Design To Implementation', in *Advanced Software and Control for Astronomy II*, edited by Alan Bridger and Nicole M. Radziwill, Proc. SPIE, Volume 7019, pp701913-1 to 701913-15.
3. Elias, J., 2009, 'Operating Concepts Definition Document', v1.9, Kosmos Project, NOAO.

4. Buchholz, N. and Daly, P. N., 2004, 'The Generic Pixel Server Dictionary', in *Advanced Software, Control and Communications Systems for Astronomy*, edited by Hilton Lewis and Gianni Raffi, Proc. SPIE, Vol. 5496, pp167-177.
5. Daly, P. N., 2010, ICD 6.1, Kosmos Project, NOAO.
6. Daly, P. N., 2010, ICD 5.1, Kosmos Project, NOAO.
7. Daly, P. N., 2010, ICD 3.1, Kosmos Project, NOAO.
8. Pogge, R. et al., 2010, ICD 3.2, Kosmos Project, NOAO.
9. Farrell, T. J., Bailey, J. A. and Shortridge, K., 1995, 'Tel/tk with DRAMA—A Natural for Building User Interfaces to Instrumentation Systems?', in *Astronomical Data Analysis Software and Systems IV*, ASP Conf. Ser., Vol. 77, edited by R. A. Shaw, H. E. Payne and J. J. E. Hayes.
10. Gillies, K., 1996, 'Programmers Guide to the Generic WIYN Client (GWC)', Mountain Programming Group, NOAO.

A Instrument Control System GWC Interface

NOCS (KOSMOS/COSMOS) INSTRUMENT COMPONENT CONTROLLER (NICC)

GWC Streams and Commands

History

v1.0.2:
- 24-Aug-2010 added camera focus, P. N. Daly (pnd@noao.edu)
v1.0.1:
- 09-Jul-2010 added corrections from S. Gott, P. N. Daly (pnd@noao.edu)
v1.0.0:
- 07-Jul-2010 original version, P. N. Daly (pnd@noao.edu)

GWC Data Streams

system stream description

kpno_4m kosmos.main main page
kpno_4m kosmos.disperser disperser wheel
kpno_4m kosmos.filter filter wheel
kpno_4m kosmos.slit slit wheel
kpno_4m kosmos.thermal thermal environment
kpno_4m kosmos.collimator collimator mechanism
kpno_4m kosmos.camera camera mechanism

ctio_4m cosmos.main main page
ctio_4m cosmos.disperser disperser wheel
ctio_4m cosmos.filter filter wheel
ctio_4m cosmos.slit slit wheel
ctio_4m cosmos.thermal thermal environment
ctio_4m cosmos.collimator collimator mechanism
ctio_4m cosmos.camera camera mechanism

Data Stream Attributes

-->kosmos.main

attribute	type	number	keyword	units	description
ident	string	1	KSIDENT	None	control program id and version
auth	string	1	KSAUTH	None	author(s) of the code
date	string	1	KSDATE	None	date of last modification
sim	integer	1	KSSIM	None	simulation mode {0 1} (1 == simulate)

-->cosmos.main

attribute	type	number	keyword	units	description
ident	string	1	CSIDENT	None	control program id and version
auth	string	1	CSAUTH	None	author(s) of the code
date	string	1	CSDATE	None	date of last modification
sim	integer	1	CSSIM	None	simulation mode {0 1} (1 == simulate)

-->kosmos.disperser

attribute	type	number	keyword	units	description
dwpos	integer	1	KSDWPOS	None	wheel actual pos {0 1 2 3 4 5 6}
dwcmd	integer	1	KSDWCMD	None	wheel demand pos {1 2 3 4 5 6}
pos	string	1	KSDPOS	None	actual name {between lo med high narrow other open}
name	string	1	KSDNAME	None	demand name {lo med high narrow other open}

-->cosmos.disperser

attribute	type	number	keyword	units	description
dwpos	integer	1	CSDWPOS	None	wheel actual pos {0 1 2 3 4 5 6}
dwcmd	integer	1	CSDWCMD	None	wheel demand pos {1 2 3 4 5 6}
pos	string	1	CSDPOS	None	actual name {between lo med high narrow other open}
name	string	1	CSDNAME	None	demand name {lo med high narrow other open}

-->kosmos.filter

attribute	type	number	keyword	units	description
fw1pos	integer	1	KSF1POS	None	wheel 1 actual pos {0 1 2 3 4 5 6}
fw1cmd	integer	1	KSF1CMD	None	wheel 1 demand pos {1 2 3 4 5 6}
fw2pos	integer	1	KSF2POS	None	wheel 2 actual pos {0 1 2 3 4 5 6}
fw2cmd	integer	1	KSF2CMD	None	wheel 2 demand pos {1 2 3 4 5 6}
pos	string	1	KSFILCMD	None	actual name {between U B V R I open}
name	string	1	KSFILTER	None	demand name {U B V R I open}

-->cosmos.filter

attribute	type	number	keyword	units	description
fw1pos	integer	1	CSF1POS	None	wheel 1 actual pos {0 1 2 3 4 5 6}
fw1cmd	integer	1	CSF1CMD	None	wheel 1 demand pos {1 2 3 4 5 6}
fw2pos	integer	1	CSF2POS	None	wheel 2 actual pos {0 1 2 3 4 5 6}
fw2cmd	integer	1	CSF2CMD	None	wheel 2 demand pos {1 2 3 4 5 6}
pos	string	1	CSFILCMD	None	actual name {between U B V R I open}
name	string	1	CSFILTER	None	demand name {U B V R I open}

-->kosmos.slit

attribute	type	number	keyword	units	description
swpos	integer	1	KSSWPOS	None	wheel actual pos {0 1 2 3 4 5 6}
swcmd	integer	1	KSSWCMD	None	wheel demand pos {1 2 3 4 5 6}
pos	string	1	KSSPOS	None	actual name {between long 1px 2px narrow other open}
name	string	1	KSSNAME	None	demand name {long 1px 2px narrow other open}

-->cosmos.slit

attribute	type	number	keyword	units	description
swpos	integer	1	CSSWPOS	None	wheel actual pos {0 1 2 3 4 5 6}
swcmd	integer	1	CSSWCMD	None	wheel demand pos {1 2 3 4 5 6}
pos	string	1	CSSPOS	None	actual name {between long 1px 2px narrow other open}
name	string	1	CSSNAME	None	demand name {long 1px 2px narrow other open}

-->kosmos.collimator

attribute	type	number	keyword	units	description
focus	float	1	KSCOLFOC	microns	collimator focus

-->cosmos.collimator

attribute	type	number	keyword	units	description
focus	float	1	CSCOLFOC	microns	collimator focus

-->kosmos.camera

attribute	type	number	keyword	units	description
focus	float	1	KSCAMFOC	microns	camera focus

-->cosmos.camera

attribute	type	number	keyword	units	description
focus	float	1	CSCAMFOC	microns	camera focus

-->kosmos.thermal

attribute	type	number	keyword	units	description
temp1	float	1	KSTEMP1	Kelvin	temperature sensor 1
temp2	float	1	KSTEMP2	Kelvin	temperature sensor 2
temp3	float	1	KSTEMP3	Kelvin	temperature sensor 3
temp4	float	1	KSTEMP4	Kelvin	temperature sensor 4

-->cosmos.thermal

attribute	type	number	keyword	units	description
temp1	float	1	CSTEMP1	Kelvin	temperature sensor 1
temp2	float	1	CSTEMP2	Kelvin	temperature sensor 2
temp3	float	1	CSTEMP3	Kelvin	temperature sensor 3
temp4	float	1	CSTEMP4	Kelvin	temperature sensor 4

GWC Commands

attribute	argument(s)	description
kosmos.main.sim	set {0 1}	simulation mode control
kosmos.filter.fw1cmd	set {1 2 3 4 5 6}	set wheel 1 to position {n}
kosmos.filter.fw2cmd	set {1 2 3 4 5 6}	set wheel 2 to position {n}
kosmos.filter.name	set {U B V R I open}	set to named filter configuration
kosmos.disperser.dwcmd	set {1 2 3 4 5 6}	set disperser wheel to position {n}
kosmos.disperser.name	set {lo med high narrow other open}	set to named disperser configuration
kosmos.slit.swcmd	set {1 2 3 4 5 6}	set slit wheel to position {n}
kosmos.slit.name	set {long 1px 2px narrow other open}	set to named slit configuration
kosmos.collimator.focus	move {focus}	move collimator to given focus
kosmos.camera.focus	move {focus}	move camera to given focus
cosmos.main.sim	set {0 1}	simulation mode control
cosmos.filter.fw1cmd	set {1 2 3 4 5 6}	set wheel 1 to position {n}
cosmos.filter.fw2cmd	set {1 2 3 4 5 6}	set wheel 2 to position {n}
cosmos.filter.name	set {U B V R I open}	set to named filter configuration
cosmos.disperser.dwcmd	set {1 2 3 4 5 6}	set disperser wheel to position {n}
cosmos.disperser.name	set {lo med high narrow other open}	set to named disperser configuration
cosmos.slit.swcmd	set {1 2 3 4 5 6}	set slit wheel to position {n}
cosmos.slit.name	set {long 1px 2px narrow other open}	set to named slit configuration
cosmos.collimator.focus	move {focus}	move collimator to given focus
cosmos.camera.focus	move {focus}	move camera to given focus

GWC Actions

ACKNOWLEDGED

BUSY

DONE

ERROR

Notes
