A guide to Nightview

Filip Hroch
Faculty of science, Masaryk University

hroch@physics.muni.cz
Kotlarska 2
Brno
611 37
Czech Republic

A guide to Nightview

by Filip Hroch

Table of Contents

List of Tables

Preface

This document describes the Nightview package, an application for control of a CCD astronomical camera along with a telescope mount.

Motivation

Our university was bought a CCD camera a few years before. The CCD acquiring software was enclosed with camera kid but only for Windows or DOS not for linux. So, we was needed a linux software for control of this camera to take a fully advantage of this very expensive instrument.

The developing of this software leaded me to a completely different view to use and control of CCD camera (or some other instrument) with a modern computer. The operation and architecture is a quite un-similar than you know from Windows versions of similar software (ccdoops etc.). I was created a operational schema to use camera in a fully network environment according to the standard Unix philosophy. This package has some advantages with respect to others programs. It implements an automatic acquiring of multi-colour image series or co-working with telescopes mount.

My first experiments with the SBIG library leaded me to sbig_exposure, sbig_filter and sbig_temperature shell routines and to a GUI front-end ccdsnaper. The ccdsnaper was a direct pre-release of the GTKnightview. The architecture of this utilities don't supports sharing of resources or any network environment. It's need set suid bit on GTK+ fronted. This (non-secure) feature is not more supported in GTK+ from 1.2 version. All of this leaded my to completely rewriting of these utilities. I was developed ccdsnaper at first quarter of 2001. The switching to a new philosophy was during summer and a new Nightview package was introduced at autumn of 2001. The developing of Nightview was coordinated with developing of xmove (The software for telescope control on *Monte Boo Observatory* (http://www.physics.muni.cz/mb/)).

The architecture of the nightview is changed since the version 0.3.x (February 2002). Now, the server is more secure and more simpler to installation (the sudo or suid are no-more required).

Chapter 1. Introduction

What is a Nightview?

The nightview is a set of a small simple utilities to a control CCD camera under Unix (linux is implemented yet). It is useful for an astronomer to study of the sky by the observation with a CCD camera equipment.

The Nightview package is low, middle and high level package to operate with CCD camera. It uses a multilayer server-client framework, therefore the clients don't depends on specific hardware, only on standard behaviour of these instruments (like acquire of image, select filter and so on). The server depends to a specific HW over its specific library, which operate on communication with CCD camera. Only SBIG camera operation is implemented yet.

Brief history

2000, Winter: Linux installed at MonteBoo
2001, Spring: first experiments with ccdsnaper GUI
2001, July: start of daemon - client framework, renamed to nightview
2001, August: start of test basic functionality of nightview
2001, October: testing completed, debianized
2001, November 15?, nightview published on freshmeat
2002, January: 0.2.x branch frozen, start on 0.3.x
2002, spring: rewrote a network engine, telescope support
2002, June: extensive testing
2002, autumn: packaged, testing finished
2004, summer: mount driver developing
2004, autumnt: xmove in python, rewrited from scratch

Feedback

Any feedback is welcome. Please, send any suggestions, notes and bugs to <*hroch@physics.muni.cz*> (mailto:hroch@physics.muni.cz).

WARRANTY

WITHOUT ANY WARRANTY. IF YOU WILL DESTROY ANY EQUIPMENT USING BY NIGHTVIEW PACKAGE IT IS YOU PROBLEM. Be careful, but the Nightview is a relative secure.

Welcome developer

A new developers are welcome. There is driver for a few kinds of the SBIG's cameras, universal driver for telescope control, but drivers for others instruments missing yet. The CookBook camera, Meade telescope mount (LXxxx) are looking for owners and programmers. Please contact me, if you have (or plan write) any another driver.

Chapter 2. The Nightview description

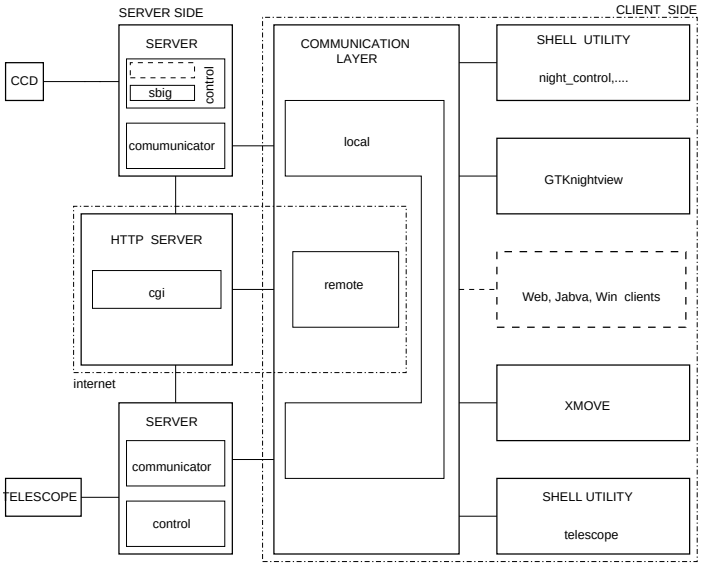
The nightview package is a server-client application. This means that, in principle, minimally two programs must run: The server program on computer with camera connected and some client program. We recommends use different computers for its due to increase efficiency during down-loading image over parallel port. Both server and client can run on the same machine, of course.

Our configuration on MonteBoo is an old P90 computer as server and (de)Celeron 433 as a platform for client. Both run the *Debian* (http://www.debian.org). The server don't need massive hardware support. It run only server and a few operational system utilities. I believe that 386 with 8Megs of memory and 50M disc will be OK, but I recommends at least 486 with respect to the higher speeds of image downloading over parallel port. The more than P90, 16M memory and 100M disc is un-warranted luxury and it don't operate faster because limited by the parallel port and net speed. The server needn't extensive use of disc space. Only the clients saves images. The server-client framework is equipped to use in TCP/IP environment. Than the utility will be work for example over intranet with ethernet, serial or radio link. The Monte Boo camera operates on one ethernet segment of the Masaryk University, so the our camera is directly connected to Internet. Note, that server part not needs some expensive libraries except libsbig, cfitsio and standard system libraries.

The nightview package contains two kinds of clients: GUI for GTK+ and shell utilities. The GTK+ front-end is usable for interactive control of the CCD. It's useful for focusing, temperature regulation or telescope pointing.

The shell client are developed for non-interactive control of CCD. They are contains all functions of the GUI package and all functions of the CCD camera. With this utilities the camera is controled over a extremely slow communication lines (telephone) and for the batch processing. There is a shell front-end - night_control - for acquiring of image series with arbitrary combination of the exposure timeouts and filters.

In principle, there is no problem for implementation of clients on other platforms or in others languages or environments. So, the windows users can connect to server with fully transparency as the linux clients for example. Moreover, I suppose that the implementation of the on-line http server with full control of the camera will be developed in the near future.



Simple block schema of nightview package.

Chapter 3. Instalation

Hardware requirements

386 processor at least (P90 recomended). About 10MB of disk space. 4MB (8MB recomended) of RAM for server or command line clients and 16MB (64MB recomended) of RAM for graphics clients.

Software requirements

The kernel version 2.2.x or better for server part (linux only). Other part will work under every modern unix. The package is developed under Debian linux distribution (glib 2.2.5, gcc 2.95.x, ..).

Downloading

The nightview package has a nice *homepage* (http://www.physics.muni.cz/mb/nightview/). Distribution site is on *integral's ftp* (ftp://integral.sci.muni.cz/pub/nightview/). The CVS server (not anonymous access) has a *web interface* (http://integral.sci.muni.cz/cgi-bin/viewcvs.cgi/nightview/) and the every day *midnight shots* (ftp://integral.sci.muni.cz/pub/nightview/) (in the nightview_YYYY_MM_DD.tar.gz file). Bugs can be send directly to me via e-mail or it can be managed over *Nightview bug database* (http://integral.sci.muni.cz/cgi-bin/gnatsweb.cgi).

Library requirement

The Nightview is a relative lightweight application. It needn't the KDE or the GNOME desktops, but it need some libraries.

Note. Please, check this *integral's ftp* (ftp://integral.sci.muni.cz/pub/nightview/) before building. The Nightview may be compiled here. If you are debian user, please use additional apt source. Add the line to your (works for stable and unstable branches) `/etc/apt/sources.list`:

```
deb ftp://integral.sci.muni.cz/debian woody main
```

.

sbig library

This is a basic low-level library for the SBIG cameras by Steve Ashe. You need download it from *Steve's ftp* (ftp://ftp.dimensional.com/users/ashe/). Please, read carefully doc enclosed in these archive. Manually install it as root:

```
# tar zxf sbig-linux-x.x.tgz                # x.x is version
# cd sbig-linux-x.x/
```

```
# cp sbig.a  /usr/local/lib/libsbig.a
# cp sbig.so /usr/local/lib/libsig.so.x.x    # rewrite version x.x
# cp sbig.h  /usr/local/include/
# cd /usr/local/lib
# ln -s libsbig.so.x.x libsbig.so.x
# ln -s libsbig.so.x.x libsbig.so
```

Now, the libsbig is prepared for use on your system. Please, be sure, that directory /usr/local/lib is in your path to search of dynamics libraries. The command

```
/sbin/ldconfig -p | grep /usr/local/lib
```

must give an output. If not (usually!), you must add this path to ld cache:

```
# editor /etc/ld.so.config       # add line /usr/local/lib
# ldconfig -v
```

If any application gives error libsbig.so.x.x not found in path..., please check this setting.

Note. I'm preparing the deb package for automatic installation. Please consult actual state on my *integral's* (ftp://integral.sci.muni.cz/pub/) I was prepared a debian package to install this library by dpkg, but this is a installer, you still needs a package's "source". Unfortunately, this installator is wrong on rpm systems (as Red Heat, Mindrake or Suse) because the rpm infrastructure don't supports the running of background down-loading for example. The ftp contains also a full version of rpm packages. I don't sure that this is absolutely correct.

Second note. The original names of this library are sbig.h, sbig.a and sbig.so. The unix "well-behaved" requires the prefix lib for an every library, so I was renamed the original name.

cFITSIO

cFITSIO (http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html) is a portable library for I/O on an astronomical image format FITS. Clients and server uses this format to save exposure, so you need it. I highly recommend use of the deb or rpm(?) package, which exists at now.

GTK, GDK, GDKpixbuf libraries

I prefer GTK graphics libraries to my development. I know that Qt, Mottif, Athena is better :-), but I prefer GTK. Sorry, if you are uses others. The GTK with GDK support and GDKpixbuf support are a relative small libraries and if you are uses gimp, than you have them. If not, they will be probably in your distribution as packages. To run of the xmove utility, the gtk-perl extension to GTK is required.

Gtk and their components are a part of your linux distribution. If not, check the *Gtk homepage* (http://www.gtk.org/) for download and instalation instructions.

The nightview in version 0.3.x was developed under gtk1.2 (1.3) library. Therefore, the sources are incompatible with a new version 2.x. The incompatibility is very deeper. It means, there is no way to compile gtk1.2 (+ gdkpixbuf) and gtk2.x with together with one source. The developers of the gtk will be support 1.2 version in future. The conclusion: you will need gtk1.2 for run of the gtknightview (and development version of one for compiling). Don't worry, both versions works nicely on one system.

I don't suppose rewriting of the graphical interface to gtk 2.x because it is really slower and it isn't equipped to run on small workstations.

cURL library

This library is used by the clients to the communication over http channel so you need it. cURL is downloadable from *cURL* (http://curl.haxx.se) but your distribution files contains probably its.

Chapter 4. Compilation

If you have all above libraries installed, you can compile Nightview:

```
i@hell:~$ cd nightview
i@hell:~$ make [-i]
```

Simply run make. The configure phase is not implemented yet, but it is probably not need because libsbig is strictly depend on PC hardware style. Now, you are waiting for binaries... The parameter -i tells make to ignore errors due to missing utilities or libraries. Use it with caution.

Problems

When the similar message appears in begin of installation:

```
..
cc -O -D_REENTRANT -I../include -L. -fPIC -c ccdsbig.c
ccdsbig.c:29: sbig.h: No such file or directory
...
```

this means that you don't installed the sbig.h header file in some system-wide include directory (do you installed the sbig library?). There is possibility use of the arbitrary directory for this file. Set the shell variable INC_SBIG to this directory. For example:

```
i@hell:~/nightview$ INC_SBIG=/home/my/sbig-linux-2.4 make
```

But this is only installation help, you still need the install libraries to run the daemon.

Chapter 5. Installation

The installation is need due to simple searching shared libraries (.so) by the binaries. If you get a messages like this:

```
night_power: error while loading shared libraries: libccdnet.so.0:
cannot open shared object file: No such file or directory
```

Your binaries are hungry for installation. Alternatively you can use setting of the environment variable LD_LIBRARY_PATH (described in Program-Library-HOWTO).

Binaries installation

Installation is normally done by typing (I'm not sure..)

```
# make install DESTDIR=/usr/local
```

It will be install packages under /usr/local directory (to bin and lib). To correct run of the utilities, the presence of the /usr/local/bin directory in your PATH environment variable and /usr/local/lib in ld config file is supposed. The binary packages (rpm and deb format) creates binaries in system wide directories.

Binary distribution

I have prepackaged debian packages for current stable distribution. The rpm packages for Red Hat linux (made under debian). There may by problem with version of rpm under various distributions (Suse has 2.x, Mandrake ?).

Documentation

The documentation is an important part of the package. It was written in the DocBook XML. So, you need installed a xml pre-processor, jade and jadetex for ps/html, pdfjadetex for pdf and xsltproc for html version of the document. Alternatively, you can use Mozilla or Opera to view this document directly as a simply structured xml. Please, ignore errors due to missing jade, jadetex, pdfjadetex, xsltproc or DocBook during compilation (the switch -i can be useful). The Nightview will be correctly work without any documentation.

You can tune some parameters for documentation. The variable JADEHTML in doc/Makefile control the html generation by jade (set to 1) or XSL preprocesor (set to 0). Try and use nicer alternative for you. Please, set additional variables (XSL and DSSSL stylesheets) to your system dependend values, if you have another of system directories than I have.

The full documentation is also available from Nightview homepage.

I've a problems with generating of mathematical formulaes from XML document. The doc directory conatins hack folder with "small hack" for its. It's the modified version of the *dbtexmath* (http://ricardo.ecn.wfu.edu/~cottrell/dbtexmath/). I'm calling for a better solution of XML wizards.

Chapter 6. Configuration

nightviewd

The main nightview server is the nightviewd. I recommends start this server at a boot time from any rc.d file. Many systems have directory /etc/init.d/ or /etc/rc.d/ where the start scripts appears. The debian users can use the prepackaged start script. If you can start nightviewd at boot time, edit some start script (/etc/init.d/local for example) and put this line to the file:

```
/usr/sbin/nightviewd
```

Command line parameters for nightviewd:

```
 -device 0x378 (default) | 0x3bc |  0x278
         lp0             | lp1   | lp2    ... IO address of the parallel port
 -filter1 x  -filter2 x ... filter5 x
         define filter specification up to 5 filters
 -site Site Name
 -longitude Longitude of the site
 -latitude  Latitude of the site
 -altitude  Altitude of the site
 -telescope Telescope designation
```

All above parameters are specified in the /etc/nightview.conf file. It's generated by nightview-conf script. This file is read first time. The command line parameters replaces the config file values.

Note. The nightviewd is not a well-matured daemon for now. It can crash every time. I highly recommends (if you have a users without root access) use some from utilities to daemon monitoring and restart it during observation. The daemon create file (socket) /tmp/.night_shock.

The nightview package contains a script nightview-conf to create the configuration file for the nightview server. The debian installation packages will be created automatically, but you can run it by the hand. The generated file has defaults used by the daemon when no configure and no command options appears. The created file /etc/nightviewd.conf follows:

```
# Config file for Nightview daemon, server part.
#
#

# ------------------------------------------------------------------------
# HW address port specification
#
# Select one from this:
#
#   port      device      address
#    1        /dev/lp0    0x378      (default)
#    2        /dev/lp1    0x3bc
#    3        /dev/lp2    0x278

Device HW address = 0x378
```

```
#
#-------------------------------------------------------------------------

#-------------------------------------------------------------------------
# Debug options
#
# Select level:
#
#   level      means
#    0         no debug informations, only status codes
#    1         print errors to standard error
#    2 - 5     from 2 to 5 increase print of debug informations
#
#    Use 0 for normal operation, increase when you have a problem.

Device HW debug level = 0

#
#-------------------------------------------------------------------------
# Site options
#
# Specify your site coordinates. This values will be used without any
# changes in header of the created FITS file.
#

# Name of site
Site = "Monte Boo"

# longitude (+east, -west) in degrees
Longitude = 16.58395

# latitude (+north, -south) in degrees
Latitude = 49.204128

# altitude (+over sea, -under sea) in meters
Altitude = 304.0

# telescope
Telescope = "0.62 m, 1:1.44 refl."

#
#-------------------------------------------------------------------------
# Filters definition
#
#  Specify string(s) for your filter(s)
#

# position in carousel = filter
Filter 1 = "B"
Filter 2 = "V"
Filter 3 = "R"
Filter 4 = "I"
Filter 5 = "clear"
```

```
#
#-------------------------------------------------------------------------
#
```

There is a possibility to change address of the connected parallel port. The switch `Device HW address = 0x378` for a first port (default) or `Device HW address = 0x3bc` for a second port in the /etc/nightview.conf file. This file provide the run-time debug level configuration also. Run `nightview-conf` script to create it. See these file for more description.

Generally, If you have the camera connected to the first parallel port, the HW address will be correct, eg. no changes are need for first tests. The new computers have two parallel ports on board, but extremely rarely are both connected to hardware connectors on a case.

The other parameters are used for fine tuning. The debug option is for hackers only (see description of debug option in sbig library). The site parameters identify your observation station. The values are no directly used by the nightview package. They're copied to the FITS header without any change.

The filter options defines your filters. The values are any strings enclosed to the apostrophes (`"`). The night_filter -list command will be print this definition in clients programs.

The running kernel needs to contain the parallel port support. Modules are preferred. Set these variables:

```
CONFIG_PARPORT=m
CONFIG_PARPORT_PC=m
CONFIG_PARPORT_PC_CML1=m
```

(this function didn't fully tested)

The server has changed logging facility from the 0.3.0 version. The syslog is used now. Any critical events (server launch, client connect, errors) will be included in syslog file. All others list from daemon are debug messages to stderr. They are enables the define the DEBUG macro in ccd.h. This logging will be disabled per default in future (in stable version).

Tip. If you have a server's computer on the net I recommends use ntpdate and ntpd to synchronise time with some time server. Include to your `/etc/crontab` this line:

```
58 * * * * root ntpdate -u 999.999.999.999 >& /dev/null &&  hwclock --systohc
```

The ntpdate will be synchronise your computer once per hour with the specified time server. The list of time servers can be found at page *Public NTP Time Servers* (http://www.eecis.udel.edu/~mills/ntp/servers.html).

Nightview server can save information about telescope's position to header of FITS file. It uses `telescope` utility to this (see mount server description). The nightview server lunch the command `telescope get -ra -dec"` to get the position and suppose as the output two real numbers. If the command is not found, output is not in fashion of two numbers or another error is appeared than the positional information is not included in the FITS header. This means that you can supply some 'telescope' utility to get info as you need. If nightviewd server run on another machine than telescoped server (and telescope utility require -host option) than use MOUNT_HOST environment variable to specify the address of the host running telescoped server.

http server

The nightview needs a http server with CGI support to run over Internet (for local clients only is not required). You can use any http server but I strongly recommends a small and a fast server like thttpd *thttpd by Jef Poskanzer* (http://www.acme.com/software/thttpd/) or boa. *boa by L. Doolittle and J. Nelson* (http://www.boa.org) Use it in standalone mode (run as own process not lunched from the inetd, moreover both boa or thttpd don't support the inetd, but wn *wn by J. Franks* (http://hopf.math.nwu.edu/) for example supports it) for a shorter response.

The scripts nightview-thttpd and nightview-boa runs the http server as system user. They are located at /ysr/sbin directory and can be used as start-up scripts. Check its and modify to fit your purposes.

If, we use the thttpd server. There is a script can be used for a lunch this server in a chroot environment in tmp directory, where the server create a log file and some temporary files.

```
# mkdir /tmp/thttpd
# ln -s /usr/bin/nightview.cgi /tmp/thttpd
# thttpd -d /tmp/thttpd -p 7666 -r -c *.cgi -l /tmp/thttpd/thttpd.log
```

The thttpd server listen on 7666 port (you can use 80 also, but the another non-nightview http server can run on this port, if you can use #80 use -host "name:80" option for clients). The all network traffic on this server will be logged to the thttp.log file.

To testing of the server use usual web browser. Copy the nightview.html file to the /tmp/thttpd directory and look at:

```
lynx http://localhost:7666/nightview.html
```

after a little bit of time appears the page:

```
This is a simple tester for nightview web interface.
If you don't know how use it, type "login" to some entry.
TELESCOPE: _____________________________________________
CAMERA: ________________________________________________
Click
...
```

Type "login" and the server send you a some message. My lynx runs the mozilla when I send the login message to the server due to sending XML back.

Testing of installation

The basic test is an output from ps. It must show the nightviewd:

```
debian:~/nightview/shell$ ps axwwu | grep nightview
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
...
root       664  0.0  0.7   2856   920 pts/1    S    19:05   0:08 ./nightviewd
f         1774  0.0  0.4   1556   540 pts/2    S    23:53   0:00 grep nightview
...
```

Ok. Nightview server runs. Than we can run the nightview-test script. It was developed to the simple tests of basic functionality of the nightview daemon and clients. Switch camera on and run it. Type to run it:

```
i@hell:~$ /usr/bin/nightview-test
```

The complete local environment (server and clients run on the same computer) will be tested. The http server can be down during this test.

If your camera works locally, you can try connect over internet. Use -host option in command line.

Chapter 7. User guide to Nightview

The GTK and shell interface for the nightview package are implemented yet. Anybody can add other clients.

Shell clients

There are a 'night_exposure', 'night_temperature', 'night_filter', 'night_power' and 'night_control' shell utilities to control exposure, temperature, filter and batch use of a CCD camera. The night_control is only a wrapper of this set of utilities to get serie of astronomical images with different exposure times and filters.

All of utilities have standard behaviour. They are driven by the command line parameters. It can be useful for scripts or single command from shell. The common options of all commands are an internet address of the server. Run its without parameters to get short description of its options. The common options -host is defaulted to file:///tmp/.night_shock, it set the communication socket to a CCD camera.

night_exposure

Night_exposure make a one snaphot with specified parameters. It is invoked with

```
night_exposure - make one exposure on the camera non-iteratively
Usage: night_exposure [-t time] [-s shutter on|off] [-b 1|2|3] [-o output]
options:
         -t    exposure time in seconds
         -s    shutter state, on - open, off - close for dark frame
         -r    select region by format: (x1,y1,x2,y2), default: full area
         -b    binning, 1 - 1x1, 2 - 2x2, 3 - 3x3, default: 3
         -c    select chip, I - imaging (default), T - tracking
         -o    name of an output file, default to nightview.fits
         -name object's name
         -obsname observer's name
         -comment some additional description enclosed in 'apostrophes'
         -z    compress output file
         -host address:port internet address server, default: local connect
Use Ctrl-C to interrupt the exposure or download in progress.
Setting of environment variable NIGHTVIEW_HOST is equivalent to -host option.
```

where -t time is exposure time in seconds (real number). -o output gives name of the output file (default: nightview.fits). The binning can be -b 1,2,3 meaning 1x1 (full resolution), 2x2 (half resolution) and 3x3 (every pixel of image is composition of 9 chip's pixels). This switch don't change the area for readout. The readout area can be changed with -r option. The -s switch control the shutter (useful for dark frames). You can specify an object name for use in 'OBJECT' keyword of the FITS file header (defaulted to no keyword will used). The switches -obsname, -comment are also useful for filling the FITS header. The option -z compress output image, so output image is saved to nightview.fits.gz by default. Use Ctrl-C to interrupt the exposure or download in progress.

The readout area is selected by four integer numbers enclosed in braces. The coordinates starts at left bottom corner according to the standard mathematical convention (eg. no stupid computer graphics con-

vention is used). The (1,1,100,100) selects a left bottom part of image. Don't forget use of apostrophes, your shell minds braces differently than you.

The switch -r (region) is useful for a faster downloading but only of part of an image. For example, it can be used for acquiring of the serie of the exposures with a fine time resolution (useful for measuring of periodic error of mount or observation of eclipses?).

The options after -o and -name requires a string. If you use a multi word name or an exotic character (please, suppose that all other a-z, A-Z and 0-9 are an exotic for your shell) enclose this string to apostrophes like this: 'F10/5.5 aperture 5 inches'.

The -o parameters specify the prefix of an output name. The resultant shape is prefix+number+character, where number designate the order in series and last character is selected filter. For example, the 10th image over R filter of the object (with prefix cluster) will be cluster10R.fits.

The shape of the output name can be changed with -oo option. The characters and so on can be arbitrary except: the integer format must precede the character and both must be presented.

Note for non-C programmers. The format "%d" will be print integers without leading zeros. The format "%02d" will be print integers with two decimal places: 01,02,03,04..09,10,11..99. Analogically for "%03d" and so on. The format "%s" will be print any string.

night_filter

Nigh filter controls a colour wheel. It's developed to set of selected filter and list of possible filters. It is invoked with

```
night_filter - select filter or print current
Usage: night_filter [-f filter] [-list]
options
         -f  set filter, one from list, may be: U,B,V,R,I
         -list list defined filters
         -host address:port internet address server, default: local connect
Setting of environment variable NIGHTVIEW_HOST is equivalent to -host option.
```

The basic action is list the possible filters with -list option. Than you can select any filter from this set by the use -f option. The current filter will be show with info option. The list is generated by nightview server so the superuser can define the names of filters in the server's configure file. See above.

night_power

Nigh power controls a basic function of camera. It switch up, down of the power and print basic info. It is invoked with

```
night_power - control the power of camera
Usage: night_power {on|off|info} [-host address]
options:
         on      switch on camera, failed if cammera not connected
         off     switch off camera
         info    print short info about camera
         -host address:port internet address server, default: local connect
```

```
Setting of environment variable NIGHTVIEW_HOST is equivalent to -host option.
```

Without parameters prints help. The 'on' connect to a camera and switch it on. The 'off' option switch of the camera to down. The 'info' option prints a basic camera info (firmware description, number of pixels, pixel's sizes and so on) if camera is on.

night_temperature

Night_temparature controls temperature regulation on a camera. It is possible to set of the checkpoint and print the current value of temperature(s) and cooling power. It is invoked with

```
night_temperatute - controls the camera temperature
Usage: night_temperature [get|set] [-t temperature] [-off] [-host address]
options
         get:
                 -t   print ccd current temperature in degrees of Celsius
                 -ta  print air current temperature in degrees of Celsius
                 -r   print current cooling power (relatively, in percents)
                 -f   print current status of fan, on or off
                  without switches print all available cooling information
         set:
                 -t temperature   set cooler temperature on value
                 -off             set temperature regulation off

         -host address:port internet address server, local connect is default
Setting of environment variable NIGHTVIEW_HOST is equivalent to -host option.
```

The basic modes of this command are an informational 'get' (print of temperature of ccd and ambient, fan status and cooling power) and 'set' (set the ccd temperature to specified temperature or set off temperature regulation). Note, it is advised to turn off regulation, but leave the fan on for a minute or so prior to shutting down the camera.

night_control

Night_control is equiped to make of series of images It is invoked with

```
night_control  -  non-interactive snapshoting with CCD camera
Usage: night_control [options] exptime1,filter1 [exptime2,filter2] ...
        -oo output filename mask, default ”
        -o output prefix name, default 'image'
        -n number of exposures, default=1
        -d dark interval, default: 0 no darks
        -w wait for specified No. of seconds, integer, default:0
        -b binning, default=3
        -name object's name
        -r select region by format: (x1,y1,x2,y2), default: full area
        -c select chip, I - imaging (default), T - tracking
        -v view exposed image with gtknightview
        -t test, only print commands, useful for background processing
        -host address:port internet address server, default: local connect
```

```
        -obsname observer's name
        -comment some additional description enclosed in 'apostrophes'
        -z compress output file
        -h print help

Setting of environment variable NIGHTVIEW_HOST is equivalent to -host option.

List of possible filters: 'B' 'V' 'R' 'I' 'clear'
```

where options are:

```
 -oo output filename mask, default 'image%d%s.fits'
 -o output prefix name, default 'image'
 -n number of exposures, default=1
 -d dark interval, default: 0 no darks
 -b binning, default=3
 -name object's name
 -r select region by format: (x1,y1,x2,y2), default: full area
 -c select chip, I - imaging (default), T - tracking
 -v view exposed image with gtknightview
 -t test, only print commands, useful for background processing
 -host address:port internet address server, default: local connect
 -obsname observer's name
 -comment some additional description enclosed in 'apostrophes'
 -z compress output file
 -h print help
```

The night_control is follows the twices of values for an exposure duration and filter for grab the serie of exposures. At least, one twice is required, but a more the one twice can be used. The full set of exposure, filter combinations will be used in one control loop.

All switches are similar to night_exposure utility. The -r, -c, -z, -name, -obsname, -comment has the same meaning as in night_exposure utility. Option -oo correspond to -o for the night_exposure.

The valid parameters are -n, -d, -v, -t. The -n represents number of exposures in series, the script will be make dark frame after specified number of images and subtract its from a scientific images with -d option. It starts before first image. The -v parameters invoke gtknightview for show a current image. The parameters and behaviour on base its can be tested with -t parameter.

Example1: The astronomer wants 100 exposures of PR Del, each 15 second in V filter, low resolution:

```
night_control -n 100 -name 'PR Del' -o 'prdel%03d%s.fits' 15,V
```

Example2: The astronomers wonts 100 exposures of PR Del, each 15 seconds on V filter, low-resolution, automatically show image after download, every 20th exposure dark:

```
night_control -v -n 100 -d 20 -name 'PR Del' -o 'prdel%03d%s.fits' 15,V
```

Example3: The astronomers wants 100 exposures of PR Del, each 15 seconds in V and 10 seconds in R filter, low-resolution, every 20th exposure dark:

```
night_control -n 100 -d 20 -name 'PR Del' -o 'prdel%03d%s.fits' 15,V 10,R
```

Technical note 1. The night_control search all binaries in standard system directories by default. If you need a run they from some non-standard place, add your bin directory to the system variable PATH. Especially, the binaries night_dark and night_keylist are required for night_control's correct run.

Environmental variables

All shell utilities (night_*) gets information from environmental variables. The command line options replaces the environmental setting in general.

NIGHTVIEW_HOST variable sets the 'host' option commonly for all shell utilities. If variable is unset this variable is defaulted to file:///tmp/.night_shock (local connection).

Example. The command

```
export NIGHTVIEW_HOST="http://my.camera.net"
```

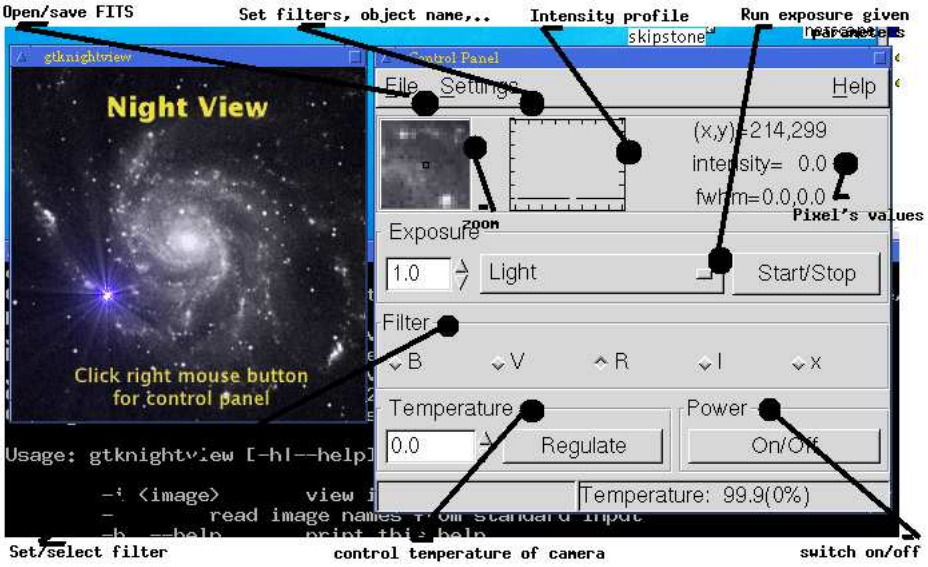
will set the host with camera conected to http://my.camera.net.

GUI interface

The GUI interface is named 'gtknightview'. You can control a CCD camera with it and view some FITS files. The interface is intuitive. It can be invoked with:

```
gtknightview [-h|--help] [-i image] [-]
```

The user can ask for help (-h option). The user can specify an image to view. The names of FITS images will read from standard input with "-" specified on command line. In this case, the nightview will be closed after end of input file. This is mode for batch processing. The nightview will work together with night_control.



Main window and control panel of nightview. The basic description is included.

The gtknightview is designed to an iterative operation with CCD instrument, not for huge series of your CCD images. Please, use the night_control utility for this purposes. I don't plan integration any series-like feature to the gtknightview's menu. It's possible that another graphics utility will be written as a frontend to night_control.

The dark image subtraction behaviour may be slightly confusing. The dark image isn't make automatically. The subtract menu item is clickable only after manually grabbed dark image. Please, check that you use a dark image with the same exposure and camera temperature.

Read-only flag on output images

In principle, every image from the CCD astronomical camera is unique. No repeating of heaven's mysteries is possible. Therefore, the utilities of nightview package creates every saved image as read-only by default. (Technically the unmask mode is set to 444 on Unix systems.) There is no switch for overwriting of this feature except patching and re-compilation of sources. If you can rewrite or delete your data (NOT RECOMMENDED), use chmod command for this purpose.

I was added this feature after unfortunately removing all of my images of the 1998WT24 asteroid, BL Lac observations and time series of SAX xxxx+xxx. Please, be careful.

Chapter 8. Mount of telescope

The nightview package also provide the server for control of telescope's mount. The shell a GTK+ client are included. They are control the telescope motions, clock and a dome slit position.

Mount's server

The server is a simpler variant of the nightview server. It listen on socket, it knows a set of commands. The server is a multi-threaded (more than one control track i one run binary). A first thread communicates with clients, a second controls a timers, start mount's engines and updates time specific variables (Julian date, azimuth of the telescope.. or spin the dome is it is need) and a third periodically saves the coordinates to file (for more robust behaviour when something crashed). The observation site parameters shares with nightview package. The server is a relative clever but it isn't the oracle. It uses all coordinates in degrees but it don't know nothing about your steeper motors. Therefore, the user should provide a few a low-level utilities (described in detail below) to do it.

Config file

The mount config file is shared with nightview daemon. It's means It read of the file /etc/nightview.conf. Only those items are used by the telescoped:

```
Longitude = 16.58395
Latitude = 49.204128
```

The geographical coordinates of your site in degrees must be provided. The longitude is coded as positive on east and negative to west with respect to zero's meridian (eg Greenwich). The latitude is in obvious manner. This values are used to computation of horizontal coordinates, Julian time etc. so enter the most precise values you knows.

Invoking options

Both latitude and longitude parameters can by independently set on command line during invoking of the daemon. The options are: -latitude and -longitude. See this example:

```
telescoped –latitude 49.204128 –longitude 16.58395
```

The command line switches overrides the config file setting. For details see the config file description.

Note. If command line switches are not specified and config file is not found, the defaults (Longitude = 16.58395, Latitude = 49.204128) are used. They are unusable for you, I suppose.

Note 2. Any part of this package wasn't tested on a wild west or a heat south observatories. Be careful.

Specification of the mount drivers

The server is independent on your hardware solution of the mount. This independence requires a site specific solution. This solution is build on a few low-level subroutines directly used to communication with mount engine.

A good example is a mount with axes driven by steeper motors. The steeper motor have 100 steps per revolution (for example), your gears provides 1:3600 ratio so the low-level subroutines may know that you have a 100 steps per degree. Moreover, this drivers may provide communication over some interface (hardware dependent again).

The nightview package defines six general commands to control of the telescope. The `telescope_rekl`, `telescope_dekl`, `telescope_clock` and `telescope_stop` controls the telescope. The `telescope_domeinit` and `telescope_dome` controls the dome. The table describe usage:

Table 8-1. Low-level commands used by telescoped

command	options
telescope_rekl	difference [deg] velocity [deg/dec]
telescope_dekl	difference [deg] velocity [deg/dec]
telescope_clock	0 | 1
telescope_stop	
telescope_domeinit	0 | 1
telescope_dome	difference [deg] velocity [deg/dec]

The `telescope_rekl` command change the Right Ascension of the telescope about value difference. If actual value of the telescope is BEGINTEXLITERALα_0ENDTEXLITERAL and the difference BEGINTEXLITERAL$\Delta \alpha$ENDTEXLITERAL than the telescope is moved to:

Equation 8-1.

BEGINTEXLITERAL$\alpha = \alpha_0 + \Delta \alpha$ENDTEXLITERAL

The difference are in degrees, positive and negative values. Second parameter is velocity of moving in degrees per second. The command should immediately print the number of seconds needs to moving as a float number. If any error is occurred the zero is printed and return value should by set to non-zero.

The `telescope_dekl` command change the Declination of the telescope about value difference. If actual value of the telescope is BEGINTEXLITERALδ_0ENDTEXLITERAL and the difference BEGINTEXLITERAL$\Delta \delta$ENDTEXLITERAL than the telescope is moved to:

Equation 8-2.

BEGINTEXLITERAL$\delta = \delta_0 + \Delta \delta$ENDTEXLITERAL

The difference are in degrees, positive and negative values. Second parameter is velocity of moving in degrees per second. The command should immediately print the number of seconds needs to moving as a float number. If any error is occurred the zero is printed and return value should by set to non-zero.

The `telescope_clock` command switch on the clock of the mount when parameter 1 (or any positive non-zero value) is supplied. If the zero (or negative or nothing) is specified, the clock is stopped. The float-number format can be used. The command should immediately print the number one (1) (may by a float number) when started and number zero (0) (may by a float number) when stopped the clocks. If any error is occurred the zero is printed and return value should by set to non-zero.

The `telescope_stop` command immediately stop all moving (clock included) motors. It's normally not used and is invoked when unconditional interrupt from user is occurred. The command line arguments are non-relevant and print the 0 (fail) or 1 (success).

The `telescope_domeinit` command initialise the dome (open the doors, spin to the telescope position for example) the 1 when parameter 1 (or any positive non-zero value) is supplied. If the zero (or negative or nothing) is specified, the dome is uninitialised (doors are closed, power is down). The float-number format can be used. The command should immediately print the number one (1) (may by a float number) when dome is initial and number zero (0) (may by a float number) when uninitialised the dome. If any error is occurred the zero is printed and return value should by set to non-zero.

The `telescope_dome` command change the azimuth of the dome's slit about value difference. If actual value of the slit is BEGINTEXLITERALA_0ENDTEXLITERAL and the difference BEGINTEXLITERALΔAENDTEXLITERAL than the dome is spin around angle:

Equation 8-3.

BEGINTEXLITERAL$A = A_0 + \Delta A$ENDTEXLITERAL

The difference are in degrees, positive and negative values. Second parameter is velocity of moving in degrees per second. The command should immediately print the number of seconds needs to moving as a float number. If any error is occurred the zero is printed and return value should by set to non-zero.

Warning. To correct behaviour of the telescoped server all listed drivers are need. Otherwise, the server will be flood your syslog with warnings and control utilities will print confusing messages. If any function is not supported by your hardware (like dome) use a dummy script. For example, telescope_dummy on my computer is:

```
#!/bin/sh
#
# dummy telescope

echo -1
exit 0
```

with corporate links.

Mount's clients

The shell and GTK+ clients are provided by the nightview package. The shell client is named `telescope` and GTK+ client is `xmove`.

telescope

The main utility to control of the telescope is `telescope`. It's command line driven. No any config files are supposed. Telescope provide setting or getting many important characteristics of the dome, telescope or date. Transparently adds support for the local or Internet connection. The on-line help is provided with invoking it without parameters:

```
Telescope mount position control 0.0.0
Use: telescope [set | get] [options, values]

         get:    (specify nothing, one or more option)
                 -ra   print Right Ascension in degrees
                 -dec  print Declination in degrees
                 -a    print Azimuth (180 deg = north)
                 -z    print Altitude
                 -ha   print Hour Angle
                 -jd   print Julian date
                 -status   print status of moving telescope

         set:    (every option needs additional parameter(s))
                 -coo RA Dec  set the Equatorial coordinates in degrees
                 -cal RA Dec calibrate the coordinates (degrees)
                 +ra   add value to Right Ascension in degrees
                 +dec  add value to Declination in degrees
                 -vra   set velocity of Right Ascension in degrees per second
                 -vdec  set velocity of Declination in degrees per seconds
                 stop    stop movine (any direction)
                 -c on   start clock-machine
                 -c off  stop clock-machine
                 -c park          park telescope

                 -host address:port Internet address of server, "localhost:7666"
                 -q don't print the telescope status during setting of telescope
```

Two main modes are supported. The get mode for getting all or specified characteristics and set mode to set and calibrate values. The key set or get should be used to identify mode. The additional parameters can follow.

Status of telescope

The get options switch to get mode. Than user can specify one, more or nothing option and required values will be print. The format of the output list is list of items, each

```
name = value
```

where name is a name (can include spaces) of the characteristics and value as a float number. If any parameter from set (jd, ra, dec...) follows the parameter as printed without leading name and a equal sign. Only float is printed.

When you specify the get option on the command line without others parameters all of characteristics will be printed:

```
debian:~/nightview/mount$ telescope get
```

```
Right Ascension =   0.000
Declination     =   0.000
Azimuth          = 156.238
Altitude        = -37.523
Hour angle      =  17.363
Dome            = 111.693
Julian date     = 2452372.3623
Siderical time  =  10.000
```

The one (or more) parameters can be added:

```
debian:~/nightview/mount$ telescope get -jd
2452372.364
```

Calibration of coordinates

The setting of various coordinates is more exciting. The coordinates are calibrated by using the -cal (abbreviation of calibrate) options follows by two float numbers meaning the current coordinates. This command print 1 on success and 0 in failure and return code is set. For example, you cantered telescope to some bright star by hand and you need tell to server when the coordinates of this star are the current. This options will be frequently used at start of observation.

```
debian:~/nightview/mount$ telescope set -cal 10 10
1
```

Setting of telescope

When the coordinates were calibrated, we can point the telescope to some interesting object with -coo option (abbreviation of coordinate) followed by the two float numbers as required coordinates again. This command print 1 on success and 0 in failure and return code is set. The motors will be switched on and telescope will be moving. The utility prints text pseudo-progress bars when telescope is moving, with switch -q this behaviour is off-ed and this routine exits immediately.

```
debian:~/nightview/mount$ telescope set -coo 11 11
1
```

The moving velocity in both axes should be separately set before any moving is executed with parameters -vra (velocity RA) and -vdec (velocity dec) followed by float numbers. This velocity is directly passed to low-level subroutines already described. This parameters should be used together with -coo option. When velocity isn't specified is set to 1 degrees per second.

```
debian:~/nightview/mount$ telescope set -vra 1 -vdec -coo 11 1
1
```

The relative difference can be specified too. Then use switches +ra and +dec (+ mean addition to current coordinates). The differences in degrees will be add to current coordinates. The moving velocity can be specified, otherwise is set to 1 deg per second. The utility prints text pseudo-progress bars when telescope is moving, with switch -q this behaviour is off-ed and this routine exits immediately.

Emergency stop

The move of telescope can be interrupted with the option stop at any time. The command can be useful when the any failure is appear. The coordinate of the telescope are undefined after this command so the new calibration is required. Practicaly, the position of the telescope will close to coordinates reported by the daemon because, the mount server continuously save a current position with a small time grid.

```
debian:~/nightview/mount$ telescope set stop
```

Clock

The switch -c followed value from a set (on, off, park) controls the clock. The option on switch up the clock, off stops the clock and park move telescope to park position (azimuth 0, height 90). This command print 1 on success and 0 in failure and return code is set.

```
debian:~/nightview/mount$ telescope set -c on
debian:~/nightview/mount$ telescope set -c off
```

This utility connect the local socket when the option -host is not used. If you can connect to a remote server than you should use the -host option followed by the Internet address of the server.

Environmental variables

The telescope utility gets information from environmental variables. The command line options replaces the environmental setting in general.

MOUNT_HOST variable sets the ’host’ option commonly for all shell utilities. If variable is unset this variable is defaulted to file:///tmp/.mount_shock (local connection).

Example. The command

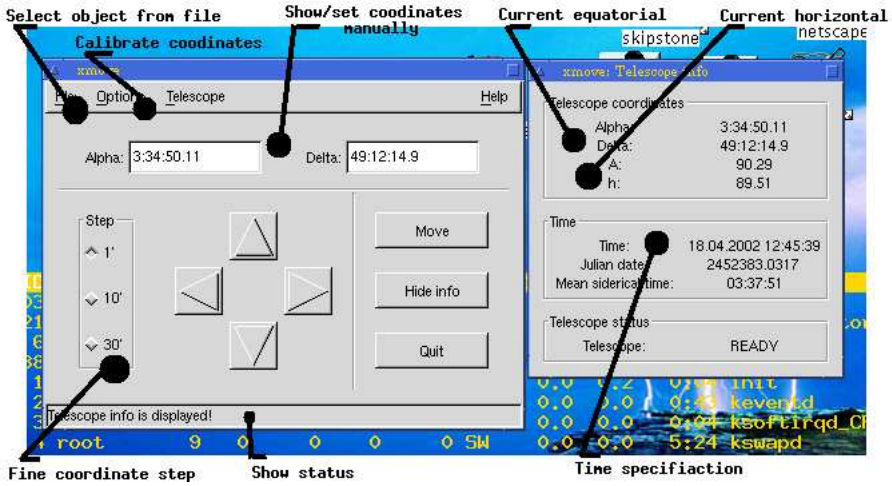
```
export MOUNT_HOST="http://my.telescope.net"
```

will set the host with camera conected to http://my.telescope.net.

xmove

Xmove is a graphical front end to the telescope utility. It requires no options. It is invoked by

```
gtknightview [-h|--help] [-i image] [-]
```



Main window and control panel of xmove. The basic description is included.

Chapter 9. FAQ

1. Hardware specific questions

Q: Do operates nightview with ST6 or older cameras connected over a serial port?

A: No. The low-level libsbig library don’t supports this feature.

Chapter 10. To Do

Test, test, test... I'm planing only fixing of bugs at near future.

Test client side on non-Linux systems.

Test the tracking chip.

Correct my English.

There is a strong need for web interface. The Java, PHP, flash programers and designers are welcome.

There is possibility to use of J.Soldan's kernel driver for SBIG cameras. This is a nice idea. The first step is done.

(Early after gtk2.0 issue) I seems the gtk2.0 isn't compatible with gtk1.2. The use another toolkit (wxWin) is possible.

Chapter 11. Bugs

The server and clients don’t supports authentification. Anybody can connect your camera when the server is running. Please, run a simple firewall on machine with camera connected.

Chapter 12. Changelog

The versions up to 0.2.3 snapshots original nightview architecture with server listening on network socket.

The version 0.3 and after implements a secure model with communication over http protocol. It implements more new features and release telescope driver utilities.

The detailed changelog can be found from integral's CVS server via *web interface.* (http://integral.sci.muni.cz/cgi-bin/viewcvs.cgi/nightview/)

Appendix A. Version of this document

$Id: nightview.xml,v 1.29 2004-12-24 16:29:37 hroch Exp $