NIRSPEC

UCLA Astrophysics Program **U.C. Berkeley** **W.M.Keck Observatory**

Fred Lacayanga **September 14, 1998**

NIRSPEC Software Programming Note 28.00
Client code file locations and descriptions

The client code resides in: **`/kroot/kss/nirspec/ui/xnirspec`**

`NirspecClientInit` : Nirspec client environment variables
`callbacks.c` : Callbacks for keyword broadcasts

`ln2_temp_callback( keyword, user_data, call_data, context )`
`scam_temp_callback( keyword, user_data, call_data, context )`
`detector_temp_callback( keyword, user_data, call_data, context )`
`track_callback( keyword, user_data, call_data, context )`
`scamguide_callback( keyword, user_data, call_data, context )`
`outdir_callback( keyword, user_data, call_data, context )`
`rootname_callback( keyword, user_data, call_data, context )`
`filesame_callback( keyword, user_data, call_data, context )`
`filenum_callback( keyword, user_data, call_data, context )`
`filename_callback( keyword, user_data, call_data, context )`
`object_callback( keyword, user_data, call_data, context )`
`comment_callback( keyword, user_data, call_data, context )`
`irot_callback( keyword, user_data, call_data, context )`
`filter_callback( keyword, user_data, call_data, context )`
`slit_callback( keyword, user_data, call_data, context )`
`echl_callback( keyword, user_data, call_data, context )`
`disp_callback( keyword, user_data, call_data, context )`
`itime_callback( keyword, user_data, call_data, context )`
`coadds_callback( keyword, user_data, call_data, context )`
`sampmode_callback( keyword, user_data, call_data, context )`
`go_callback( keyword, user_data, call_data, context )`
`test_callback( keyword, user_data, call_data, context )`
`abort_callback( keyword, user_data, call_data, context )`
`framerdy_callback( keyword, user_data, call_data, context )`
`frameend_callback( keyword, user_data, call_data, context )`
`outdir2_callback( keyword, user_data, call_data, context )`
`rootname2_callback( keyword, user_data, call_data, context )`
`filesame2_callback( keyword, user_data, call_data, context )`
`filenum2_callback( keyword, user_data, call_data, context )`
`filename2_callback( keyword, user_data, call_data, context )`
`comment2_callback( keyword, user_data, call_data, context )`
`itime2_callback( keyword, user_data, call_data, context )`
`coadds2_callback( keyword, user_data, call_data, context )`
`sampmode2_callback( keyword, user_data, call_data, context )`
`go2_callback( keyword, user_data, call_data, context )`
`test2_callback( keyword, user_data, call_data, context )`
`abort2_callback( keyword, user_data, call_data, context )`
`framerdy2_callback( keyword, user_data, call_data, context )`
`frameend2_callback( keyword, user_data, call_data, context )`
`tspstat_callback( keyword, user_data, call_data, context )`

`cli.c` : Routines that provide a command line user interface using Tcl as the command interpreter.

cli_server.c : CLI server access routines
cnirspec*
create_interest.c : Create keyword interest
dataviews.c : Client GUI routines

DV_init(void) : Initialize Data Views environment.
Link Data Views variables to client program variables
Initialize display windows
Initialize sub views
Initialize input objects

DV_handle(void) : Handle Data Views events.
Poll for specific events
On button press, go to the ***handle_button_press( )*** function
Update sliders and observing parameters

DV_expStatus(int chan, int itime_left, int itime_done) : Update exposure status display.

DV_updateINS(void) : Update instrument control window

DV_close(void) : Close all Data Views displays.

create_screen(char *devname, int screen_num) : Create a window and load a view into it.

drawport_init(char *view_name) : Initialize drawport
Load the view from the file
Bind Data Views variables to client program variables
Draw contents of the drawport

drawport_new(char *view_name) : create a new drawport

vdps_init(void) : initialize client variables that are bound to Data Views variables with default values.

vdps_rebind(OBJECT data_obj, VARDESC vdp, ADDRESS argblock) : Bind Data Views variable to client program variable.

input_objects_init(int index) : Initialize input object components and post a service result request.
The service result request looks at the `input_objects[ ]` array to determine the callback function that is automatically invoked when input is received from the input object.

popup_draw(int index, DRAWPORT drawport, char erase) : add a pop up menu to the active drawport's view.

popup_delete(int index) : delete pop up menu

popup_deleteAll(void): delete all pop up menus

```
menu_inst_setup_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```
: callback function for the pop up menu for the setup button in the instrument display window.
Checks the value in `VdpBuf_btn_inst_setup` returned from the `menu_inst_setup` pop up menu.
1 : Saves configuration file
2 : Reads configuration file
4 : Reset CLI
5 : Redraw all windows

```
menu_inst_eng_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```
Check value in `VdpBuf_btn_inst_eng` returned from the `menu_inst_eng` pop up menu.

1 : Locks engineering menu
2 : Motor pop up menu
3 : TSPLINK pop up menu
4 : Clock pop up menu
5 : Offsets pop up menu
6 : Lists nirspec clients

menu_inst_eng_motor_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_inst_eng_motor` returned from the `menu_inst_eng_motor` pop up menu

1 : Init motors
2 : Move motors
3 : Read motor position
4 : Set motor position
5 : Read motor switches

menu_inst_eng_motor_init(void)

Draw motor initialization sub view
Input loop similar to `handle_button_press( )`
Waits for Abort, Init, or Dismiss button press.

init_motors(void) : Sends appropriate initialization keyword to the server

menu_inst_eng_motor_move(void)

Draw motor move sub view
Input loop similar to `handle_button_press( )`
Waits for Abort, Init, or Dismiss button press.

move_motors(void) : Sends appropriate motor move keyword to the server

abort_motors(void) : Sends the appropriate motor abort keyword to the server

menu_inst_eng_motor_readpos(void)

Sends tspupdate to read motor positions
Sends "loc" keyword for all motors to the server
Draws readpos subview

menu_inst_eng_motor_setpos(void)

Draw motor set position sub view
Input loop similar to `handle_button_press( )`
Waits for Set or Dismiss button press.

set_motors(void) : Send the appropriate motor initloc keyword to the server

menu_inst_eng_motor_readsw(void)

Sends tspupdate to read motor switches
Sends "sw" keyword for all motors to the server
Decodes switch bits 0-3 for each motor (1 = off, 0 = on)
Draws readsw subview

menu_inst_eng_tsplink_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_inst_eng_tsplink` returned from the `menu_inst_eng_tsplink` pop up menu.

1 : Write menu
2 : Sends tsptrace keyword to the server
3 : Resets TSP link (not implimented)

menu_inst_eng_tsplink_write(void)

Draw tsplink write sub view

Input loop similar to `handle_button_press( )`

Waits for Send or Dismiss button press.

Sends cid and param to transputer

```
menu_inst_eng_clock_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Checks value in `VdpBuf_btn_inst_eng_clock` returned from the `menu_inst_eng_clock` pop up menu.

1 : Spectrometer menu

2 : SCAM menu

```
menu_inst_eng_clock_spec_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Checks value in `VdpBuf_btn_inst_eng_clock_spec` returned from the `menu_inst_eng_clock`_spec pop up menu.

1 : Frame test

2 : Fifo test

```
menu_inst_eng_clock_scam_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Checks value in `VdpBuf_btn_inst_eng_clock_scam` returned from the `menu_inst_eng_clock_scam` pop up menu.

1 : Frame test

2 : Fifo test

frame_test(int channel)

Draw iteration subview

Send frame.test keyword with the number of iterations to the server

fifo_test(int channel)

Draw iteration subview

Send fifo.test keyword with the number of iterations to the server

```
menu_inst_eng_offsets_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Checks value in `VdpBuf_btn_inst_eng_offsets` returned from the `menu_inst_eng_offsets` pop up menu.

1 : Spectrometer offset subview

2 : SCAM offset subview

menu_inst_eng_offsets_spec(void)

Draw spectrometer offset subview

Input loop similar to `handle_button_press( )`

Waits for Set or Dismiss button press.

menu_inst_eng_offsets_scam(void)

Draw SCAM offset subview

Input loop similar to `handle_button_press( )`

Waits for Set or Dismiss button press.

set_offsets(int channel) : Sends spectrometer or scam offsets to the server

```
menu_inst_help_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```
: Draws help menu

```
menu_inst_irot_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Checks value in `VdpBuf_btn_inst_irot` returned from the `menu_inst_irot` pop up menu.

1 : Turn tracking on and start watch_imrot program

2 : Turn tracking off and stop watch_imrot program

```
menu_inst_fil1_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Check filter wheel initialization status

Check value in `VdpBuf_btn_inst_filter` returned from the `menu_inst_filter` pop up menu.

2 – ll : Set filter position

12 - 22 : Blocker menu

SetFilterPos(int index, int FilterWheel, int FilterPosIndex) : Send fil1pos or fil2pos keyword to the server

```
menu_inst_fil2_1_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Checks value in `VdpBuf_btn_inst_fil2_1` returned from the `menu_inst_fil2_1` pop up menu.

1 : Set blocker

2 : Set blocker

```
menu_inst_slitview_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
    ADDRESS             buffer )
```

Checks value in `VdpBuf_btn_inst_slitview` returned from the `menu_inst_slitview` pop up menu.

1 : Turn on SCAM guiding

2 : Turn off SCAM guiding

```
menu_inst_slit_callback(
    OBJECT              index,
    EVENT_REQUEST       request,
    INT                 action,
    OBJECT              loc_event,
```

ADDRESS buffer)

Check slit wheel initialization status

Checks value in `VdpBuf_btn_inst_slit` returned from the `menu_inst_slit` pop up menu.

Sends slitpos keyword to the server

menu_spec_setup_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_spec_setup` returned from the `menu_spec_setup` pop up menu.

1 : Draw observing setup subview and update observing setup

menu_spec_file_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_spec_file` returned from the `menu_spec_file` pop up menu.

1 : Save test frame

2 : Toggle file overwrite

menu_spec_script_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_spec_script` returned from the `menu_spec_script` pop up menu.

1 : Run script

2 : Check script

3 : Edit script

4 : Abort script

menu_spec_sampmode_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_spec_sampmode` returned from the `menu_spec_sampmode` pop up menu.

menu_scam_setup_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_scam_setup` returned from the `menu_spec_scam` pop up menu.

1 : Draw observing setup subview and update observing setup

menu_scam_file_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_scam_file` returned from the `menu_scam_file` pop up menu.

1 : Save test frame

2 : Toggle file overwrite

menu_scam_sampmode_callback(
OBJECT index,
EVENT_REQUEST request,
INT action,
OBJECT loc_event,
ADDRESS buffer)

Checks value in `VdpBuf_btn_scam_sampmode` returned from the `menu_scam_sampmode` pop up menu.

handle_button_press(OBJECT location) : Handle all button press events

Check if right mouse button was pressed. This is the menu cancel action.

If the button press is not a cancel, then check the object name of the region under the cursor where the button press occurred.

Valid object names:

`btn_inst_setup`	:	Setup popup menu
`btn_inst_eng`	:	Engineering popup menu
`btn_inst_help`	:	Help menu
`btn_inst_quit`	:	Quit client
`btn_inst_lamp`	:	Calibration Lamps
`btn_inst_irot`	:	Image rotator popup menu
`btn_inst_filter`	:	Filter popup menu
`btn_inst_slitview`	:	SCAM guiding popup menu
`btn_inst_slit`	:	Slit popup menu
`btn_inst_resmode`	:	Resolution mode popup menu
`btn_inst_echelle`	:	Echelle menu
`btn_inst_temps`	:	Temperature subview
`btn_spec_setup`	:	Spectrometer observing setup popup menu
`btn_spec_file`	:	File popup menu
`btn_spec_script`	:	Script popup menu
`btn_spec_sampmode`	:	Sampling mode popup menu
`btn_spec_go`	:	Write all observing parameters and start integration
`btn_spec_test`	:	Take a test frame
`btn_spec_abort`	:	Abort exposure
`btn_scam_setup`	:	SCAM observing setup popup menu
`btn_scam_file`	:	File popup menu
`btn_scam_sky`	:	Take a sky frame
`btn_scam_script`	:	Script popup menu
`btn_scam_sampmode`	:	Sampling mode popup menu
`btn_scam_go`	:	Write all observing paramenters and start integration
`btn_scam_test`	:	Take a test frame
`btn_scam_abort`	:	Abort exposure

get_expstatus_objects(void) : load exposure status objects

subview_load(char *viewname, OBJECT screen, DISPLAY_INFO *disp_info, char stretch) : Load a view and create a drawport

subview_draw(int index, char event_poll)

Draw subview

Input loop similar to `handle_button_press( )`

Waits for OK or cancel button press.

subview_erase(int index) : Erase subview and redraw region.

update_setup(int chan)

Update observing parameters:

Telescope

Observer

Datapath

Rootname
Filenum

slider_itime(char increment) : update integration time slider object

slider_coadds(char increment) : update coadds slider object

update_obsparam(int chan)

Update observing parameters:

Object
Comment
Filter
Itime
Coadds
Sampmode

get_automsg(char *message, int seconds, int screen) : display a message

get_message(char *message, int screen) : display a single line message

get_mmessage(char *title, char *message, int screen) : display a multi line message

get_confirm(char *title, char *message, int screen) : Get confirmation

get_confirm2(char *title, char *message, int screen) : Get confirmation

get_input(char *title, char *name, char *string, int screen, int echo) : Get input parameter

get_input2(char *title, char *name, char *name2, char *string, char *string2, int screen) : Get 2 input parameters

check_access(void) : Check access to instrument control window

DV_updateScreen(int screen) : Redraw screen

DV_updateCurrentObs(int chan, int test_flag) : update observing parameters

wheel_move(char *object_name) : Move filter or slit wheel dynamic object

dataviews.h

DATA_INFO structure
POPUP_INFO structure
DISPAY_INFO structure
DIALOG_INFO structure
Dialog subview indexes
Dialog subview table
Input object indexes
input_object table
Data table

dcs_util.c : Routines that handle DCS functions. Not used.

drawings/ : Draw files (*.dr) for Data Views

efs_client.c : Routines that handle socket communications with EFS. Not used

efs_client.so

efs_gateway

efs_gateway.c : EFS gateway main function and callbacks. Not used.

efs_server.c : Routines that handle socket communications with EFS. Not used.

efs_test/

ftp.sh*

layouts/ : Layout files (*.lay) for Data Views

makefile

makefile_purify

misc.c: exposure control and other miscellaneous routines

init_globalvars(void) : Initialize observing parameters

exp_start(int chan, int test_flag) : Start integration and update status displays

exp_abort(int chan)

exp_end(int chan)

exp_done(int chan)

exp_updateStatus(int chan) : Calculate remaining integration time and update status displays

write_obsParam(int chan, int test_flag) : Send observing parameter keywords to the server.

write_abort(int chan) : send abort keyword to the server

alarmHandler(void)

Filename_get(int chan, char *filename) : Build filename from rootname and file number.

Filename_make(int chan, char *filename) : Increment the file number and write filenum keyword to the server.

get_hostname(char *host)

config_save(char *filename) : Save configuration file. Write only the keywords that are both readable and writable.

config_read(char *filename)

STP_getPosIndex(int motor_index, char *pos_name) : get motor position index according to the motor tables in nirspec.h

STP_getFilterPosIndex(char *pos_name) : get filter position index according to the filter table in nirspec.h

lookup(char *keyword) : get keyword index in KeywordTable[ ] array.

nirspec.h -> ../../keyword/nirspec.h*

old/

purify_cache/

ql_client.c

ql_server.c

run_efs*

run_nirspec_client*

run_ql*

run_rql*

run_sefs*

run_sql*

run_temp*

scripts/

socket.c : Socket client and server routines that handle socket communications with the GUI

views/ : View files (*.v) for Data Views

xnirspec*

xnirspec.c

main(int argc, char *argv[])

Check arguments

Set up error logging

Create Nirspec interest

Create DCS interest

Loop to process X, CLI and KTL events

Create a fd set consisiting of the X socket fd, CLI socket fd and KTL fd

Block until an X, CLI, or KTL event arrives

Handle X events

Get command entry from CLI

Get input from QL

Invoke KTL event handler: KTL_DISPATCH(khand);

Quit the program

xnirspec.h

Engineering password

Keyword read/write macros

```
READ_INT_KEYWORD( hand, keyword, value )
READ_DBL_KEYWORD( hand, keyword, value )
READ_STR_KEYWORD( hand, keyword, value )
WRITE_INT_KEYWORD( hand, keyword, value )
WRITE_DBL_KEYWORD( hand, keyword, value )
WRITE_STR_KEYWORD( hand, keyword, value )
```

Error macro

ERROR(msg)

Program variables that link to Data Views variables

Data structures

EXP_PARAM

OBS_SETUP

xresources/