Contents

Classes

SDIAutoParallel

Inherits from: **XDIBASE**
Class Data:

(*long*)	id	(*string*)	status	(*float*)	wavelength
(*double*)	start_time	(*float*)	param	(*int*)	step
(*int*)	nominal	(*int*)	leg1	(*int*)	leg2
(*int*)	leg3	(*int*)	curr_leg	(*int*)	param_pos
(*ptr*)	ref_image	(*int*)	get_ref_flag	(*string*)	obj_num
(*structure*)	geometry	(*int*)	need_frame	(*int*)	need_timer
(*int*)	auto	(*structure*)	palette	(*obj*)	manager

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdiautoparallel__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

data=data: No Doc

restore_struc=restore_struc: No Doc

Example Call:

$$result = \textbf{SDIAutoParallel->}\ \textbf{init}(data = data,\ restore\_struc = restore\_struc)$$

(pro) CLEANUP

Method Documentation:
No Doc
Arguments:

log: No Doc

Example Call:

SDIAutoParallel–> cleanup, *log*

(pro) FRAME_EVENT

Method Documentation:
No Doc
Arguments:

image: No Doc

channel: No Doc

Example Call:

SDIAutoParallel–> frame_event, *image*,
channel

(function) GET_SETTINGS

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **SDIAutoParallel–> get_settings**()

(pro) START_PARALLEL

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIAutoParallel–> start_parallel, *event*

(pro) STOP_PARALLEL

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIAutoParallel–> stop_parallel, *event*

SDIEtalonScanner

Inherits from: **XDIBASE**

Class Data:

(*long*)	id	(*string*)	status	(*float*)	wavelength
(*double*)	start_time	(*int*)	nchann	(*string*)	obj_num
(*structure*)	geometry	(*int*)	need_frame	(*int*)	need_timer
(*int*)	auto	(*structure*)	palette	(*obj*)	manager

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdietalonscanner__define.pro

---

METHODS:

(function) INIT

Method Documentation:
Initialize the EtalonScanner.
Arguments:

data=data: Misc data

restore_struc=restore_struc: Restored settings

Example Call:

$$result = \textbf{SDIEtalonScanner->}\ \textbf{init}(data = data,\ restore\_struc = restore\_struc)$$

(pro) CLEANUP

Method Documentation:
Cleanup, stop any current scans.
Arguments:

log: No Doc

Example Call:

$$\textbf{SDIEtalonScanner->}\ \textbf{cleanup}, log$$

(pro) FRAME_EVENT

Method Documentation:
A new frame has been recieved. Update leg diagrams, decide if we need to start a new scan.
Arguments:

image: The new camera frame

channel: The current scan channel

Example Call:

SDIEtalonScanner–> frame_event, *image*, *channel*

(function) GET_SETTINGS

Method Documentation:
Select settings to save.
Takes no arguments
Example Call:

result = **SDIEtalonScanner–> get_settings**()

(pro) PAUSE_SCAN

Method Documentation:
Pause the current scan.
Arguments:

event: Widget event

Example Call:

SDIEtalonScanner–> pause_scan, *event*

(pro) SET_WAVELENGTH

Method Documentation:
Set the wavelength for scanning.
Arguments:

event: Widget event

Example Call:

SDIEtalonScanner–> set_wavelength, *event*

(pro) START_SCAN

Method Documentation:
Start a scan.
Arguments:

event: Widget event

Example Call:

SDIEtalonScanner–> start_scan, *event*

(pro) STOP_SCAN

Method Documentation:
Stop the current scan (will restart from beginning on next 'start')
Arguments:

event: Widget event

Example Call:

SDIEtalonScanner–> stop_scan, *event*

SDIEtalonSpacer

Inherits from: **XDIBASE**

Class Data:

(*long*)	id	(*string*)	status	(*int*)	step
(*int*)	leg1	(*int*)	leg2	(*int*)	leg3
(*string*)	obj_num	(*structure*)	geometry	(*int*)	need_frame
(*int*)	need_timer	(*int*)	auto	(*structure*)	palette

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdietalonspacer__define.pro

---

METHODS:

(function) INIT

Method Documentation:
EtalonSpacer initialization.
Arguments:

data=data: Misc data

restore_struc=restore_struc: Restored settings

Example Call:

$$result = \textbf{SDIEtalonSpacer->}\ \textbf{init}(data = data,\ restore\_struc = restore\_struc)$$

(pro) ADJUST_LEGS_EVENT

Method Documentation:
An event from the widget sloders representing leg voltages.
Arguments:

event: Widget event

Example Call:

$$\textbf{SDIEtalonSpacer->}\ \textbf{adjust\_legs\_event}, event$$

(pro) CLEANUP

Method Documentation:
Cleanup - nothing to do
Arguments:

log: No Doc

Example Call:

SDIEtalonSpacer–> cleanup, *log*

(function) GET_SETTINGS

Method Documentation:
Get settings for saving.
Takes no arguments
Example Call:

result = **SDIEtalonSpacer–> get_settings**()

(pro) STEP_CHANGE

Method Documentation:
Change the size of the tilt adjustment.
Arguments:

event: Widget event

Example Call:

SDIEtalonSpacer–> step_change, *event*

(pro) TILT

Method Documentation:
A tilt event, for adjusting along the two orthogonal axes.
Arguments:

event: Widget event

Example Call:

SDIEtalonSpacer–> tilt, *event*

SDIPhaseMapper

Inherits from: **XDIBASE**

Class Data:

(*long*)	id	(*int*)	nscans	(*int*)	current_scan
(*int*)	scanning	(*int*)	nchann	(*float*)	wavelength
(*int*)	channel	(*ptr*)	image	(*ptr*)	phasemap
(*int*)	xdim	(*int*)	ydim	(*ptr*)	p
(*ptr*)	q	(*ptr*)	px	(*ptr*)	qx
(*int*)	source_order	(*float*)	source_lambda	(*ptr*)	source_pmap
(*int*)	current_source	(*float*)	gain	(*float*)	exptime
(*float*)	smooth_window	(*string*)	obj_num	(*structure*)	geometry
(*int*)	need_frame	(*int*)	need_timer	(*int*)	auto
(*structure*)	palette	(*obj*)	manager	(*obj*)	console

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdiphasemapper__define.pro

---

METHODS:

(function) INIT

Method Documentation:
Phasemapper initialization.
Arguments:

restore_struc=restore_struc: Misc data

data=data: Restored settings

Example Call:

$$result = \textbf{SDIPhaseMapper}\text{-}\!> \textbf{init}(restore\_struc = restore\_struc, \\ data = data)$$

(function) AUTO_START

Method Documentation:
Auto start the Phasemapper - called whn running in auto mode, and plugin is started from a scheduled command.
Arguments:

args: String of arguments passed from the schedule file

Example Call:

$$result = \textbf{SDIPhaseMapper}\text{-}\!> \textbf{auto\_start}(args)$$

(pro) CLEANUP

Method Documentation:
Cleanup, close any active scans.
Arguments:

log: No Doc

Example Call:

SDIPhaseMapper–> cleanup, *log*

(pro) FRAME_EVENT

Method Documentation:
Frame event - update the Fourier summations for every pixel, if scan is finished, finalize and unwrap the phasemap, and save it.
Arguments:

image: Latest frame from the camera

channel: Current scan channel

Example Call:

SDIPhaseMapper–> frame_event, *image*,
channel

(function) GET_SETTINGS

Method Documentation:
Get settings to save.
Takes no arguments
Example Call:

result = **SDIPhaseMapper–> get_settings**()

(pro) SET_INTERP

Method Documentation:
When using more than one wavelength to generate a phasemap, we set the order of the cal sources (the numbers corresponding to positions of the calibration source selector switch) and the wavelengths those sources correspond to. The info from both phasemaps is store in such a way as to allow the spectral plugin to interpolate between the phasemaps at the two wavelengths.
Arguments:

event: Widget event

Example Call:

SDIPhaseMapper–> set_interp, *event*

(pro) SET_NUM_SCANS

Method Documentation:
Set the number of scans to co-add.
Arguments:

event: Widget event

Example Call:

SDIPhaseMapper–> set_num_scans, *event*

(pro) SET_SMOOTH_WINDOW

Method Documentation:
Set the width of the smoothing window, applied after phasemap is unwrapped.
Arguments:

event: Widget event

Example Call:

SDIPhaseMapper–> set_smooth_window, *event*

(pro) START_SCAN

Method Documentation:
Start scanning.
Arguments:

event: Widget event

Example Call:

SDIPhaseMapper–> start_scan, *event*

(pro) STOP_SCAN

Method Documentation:
Stop the current scan.
Arguments:

event: Widget event

Example Call:

SDIPhaseMapper–> stop_scan, *event*

SDISharpness

Inherits from: **XDIBASE**

Class Data:

(*long*)	id	(*float*)	sbuffer	(*float*)	history
(*int*)	count	(*int*)	bcount	(*float*)	best
(*int*)	leg1_best	(*int*)	leg2_best	(*int*)	leg3_best
(*int*)	xcen	(*int*)	ycen	(*int*)	xdim
(*int*)	ydim	(*string*)	obj_num	(*structure*)	geometry
(*int*)	need_frame	(*int*)	need_timer	(*int*)	auto
(*structure*)	palette	(*obj*)	manager	(*obj*)	console

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdisharpness__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

restore_struc=restore_struc: No Doc

data=data: No Doc

Example Call:

$$result = \textbf{SDISharpness->}\ \textbf{init}(restore\_struc = restore\_struc,\ data = data)$$

(pro) CLEANUP

Method Documentation:
No Doc
Arguments:

log: No Doc

Example Call:

SDISharpness-> **cleanup**, *log*

(pro) FRAME_EVENT

Method Documentation:
No Doc
Arguments:

image: No Doc

channel: No Doc

scan: No Doc

Example Call:

SDISharpness–> frame_event, *image*,
channel,
scan

(pro) GET_CENTER

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDISharpness–> get_center, *event*

(function) GET_SETTINGS

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **SDISharpness–> get_settings**()

SDISpectrum

Inherits from: **XDIBASE**

Class Data:

(*long*)	id	(*int*)	scanning	(*int*)	nchann
(*int*)	xdim	(*int*)	ydim	(*int*)	save_file_id
(*ptr*)	spectra	(*ptr*)	last_spectra	(*ptr*)	zonemap
(*ptr*)	zonemap_boundaries	(*ptr*)	phasemap	(*float*)	signal_noise_history
(*float*)	channel_background_history	(*float*)	scan_background_history	(*ptr*)	zone_centers
(*int*)	nzones	(*string*)	dll	(*int*)	nscans
(*int*)	file_id	(*string*)	zone_settings	(*float*)	wavelength
(*float*)	a	(*float*)	b	(*float*)	c
(*double*)	scan_start_time	(*string*)	spec_path	(*int*)	nrings
(*string*)	file_name_format	(*string*)	filename	(*ptr*)	rads
(*ptr*)	secs	(*ptr*)	accumulated_image	(*int*)	finalize_flag
(*string*)	insprof_filename	(*float*)	etalon_gap	(*string*)	obj_num
(*structure*)	geometry	(*int*)	need_frame	(*int*)	need_timer
(*int*)	auto	(*structure*)	palette	(*obj*)	manager

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdispectrum__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

restore_struc=restore_struc: No Doc

data=data: No Doc

zone_settings=zone_settings: No Doc

file_name_format=file_name_format: No Doc

Example Call:

$$result = \textbf{SDISpectrum}\text{->}\ \textbf{init}(restore\_struc = restore\_struc,\ data = data,\ zone\_settings = zone\_settings,\ file\_name\_format = file\_name\_format)$$

(function) AUTO_START

Method Documentation:
No Doc
Arguments:

args: No Doc

Example Call:

$$result = \textbf{SDISpectrum->}\ \textbf{auto\_start}(args)$$

(pro) CLEANUP

Method Documentation:
No Doc
Arguments:
log: No Doc

Example Call:

SDISpectrum-> cleanup, *log*

(pro) FINALIZE_SCAN

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

SDISpectrum-> finalize_scan, *event*

(pro) FIT_SPECTRA

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

SDISpectrum-> fit_spectra, *event*

(pro) FRAME_EVENT

Method Documentation:
No Doc
Arguments:

image: No Doc

channel: No Doc

Example Call:

SDISpectrum–> frame_event, *image*,
channel

(function) GET_SETTINGS

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **SDISpectrum–> get_settings**()

(pro) INITIALIZER

Method Documentation:
No Doc
Takes no arguments
Example Call:

SDISpectrum–> initializer

(pro) SET_PHASEMAP

Method Documentation:
No Doc
Arguments:

failed: No Doc

Example Call:

SDISpectrum–> set_phasemap, *failed*

(pro) START_SCAN

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDISpectrum–> start_scan, *event*

(pro) STOP_SCAN

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDISpectrum–> stop_scan, *event*

SDIStepsPerOrder

Inherits from: **XDIBASE**

Class Data:

(*long*)	id	(*ptr*)	corr	(*int*)	num_chords
(*int*)	curr_chord	(*int*)	scanning	(*int*)	nchann
(*int*)	start_volt_offset	(*int*)	stop_volt_offset	(*float*)	volt_step_size
(*obj*)	scan_obj	(*int*)	curr_chann	(*int*)	last_chann
(*ptr*)	image	(*ptr*)	ref_image	(*int*)	xdim
(*int*)	ydim	(*int*)	counter	(*int*)	last_counter
(*ptr*)	chord_hist	(*float*)	wavelength	(*int*)	record_value
(*string*)	record_file	(*float*)	gain	(*float*)	exptime
(*string*)	obj_num	(*structure*)	geometry	(*int*)	need_frame
(*int*)	need_timer	(*int*)	auto	(*structure*)	palette

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdistepsperorder__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

restore_struc=restore_struc: No Doc

data=data: No Doc

Example Call:

$$result = \textbf{SDIStepsPerOrder->}\ \textbf{init}(restore\_struc = restore\_struc,\ data = data)$$

(function) AUTO_START

Method Documentation:
No Doc
Arguments:

args: No Doc

Example Call:

$$result = \textbf{SDIStepsPerOrder->}\ \textbf{auto\_start}(args)$$

(pro) CLEANUP

Method Documentation:
No Doc
Arguments:
log: No Doc

Example Call:

SDIStepsPerOrder–> cleanup, *log*

(pro) FRAME_EVENT

Method Documentation:
No Doc
Arguments:
image: No Doc
channel: No Doc

Example Call:

SDIStepsPerOrder–> frame_event, *image*,
channel

(function) GET_SETTINGS

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **SDIStepsPerOrder–> get_settings**()

(pro) START_SCAN

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

SDIStepsPerOrder–> start_scan, *event*

(pro) STOP_SCAN

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIStepsPerOrder–> stop_scan, *event*

(pro) TOGGLE_RECORD

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIStepsPerOrder–> Toggle_Record, *event*

SDIVidshow

Inherits from: **XDIBASE**

Class Data:

(*long*)	id	(*int*)	inst	(*float*)	exp_time
(*int*)	xdim	(*int*)	ydim	(*int*)	scale
(*float*)	scale_fac	(*int*)	crosshairs	(*int*)	crosshairs_point
(*int*)	grid	(*int*)	color_table	(*long*)	framecount
(*double*)	tstrt	(*int*)	mask_quadrants	(*string*)	obj_num
(*structure*)	geometry	(*int*)	need_frame	(*int*)	need_timer
(*int*)	auto	(*structure*)	palette	(*obj*)	manager

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/sdividshow__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

restore_struc=restore_struc: No Doc

data=data: No Doc

Example Call:

$$result = \textbf{SDIVidshow}\text{->}\ \textbf{init}(restore\_struc = restore\_struc,\ data = data)$$

(pro) CLEANUP

Method Documentation:
No Doc
Arguments:

log: No Doc

Example Call:

SDIVidshow–> **cleanup**, *log*

(pro) FIT_WINDOW

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

SDIVidshow–> **fit_window**, *event*

(pro) FRAME_EVENT

Method Documentation:
No Doc
Arguments:
image: No Doc
channel: No Doc

Example Call:

SDIVidshow–> **frame_event**, *image*,
channel

(function) GET_SETTINGS

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **SDIVidshow**–> **get_settings**()

(pro) MASK_QUADRANTS

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

SDIVidshow–> **mask_quadrants**, *event*

(pro) SCALING

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIVidshow–> scaling, *event*

(pro) SET_COLOR_TABLE

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIVidshow–> set_color_table, *event*

(pro) SET_CROSSHAIRS

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIVidshow–> set_crosshairs, *event*

(pro) SET_CROSSHAIRS_POINT

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIVidshow–> set_crosshairs_point, *event*

(pro) SET_GRID

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIVidshow–> set_grid, *event*

(pro) SET_SCALE

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

SDIVidshow–> set_scale, *event*

XDIBase

Inherits from: **None**

Class Data:

(*string*)	obj_num	(*structure*)	geometry	(*int*)	need_frame
(*int*)	need_timer	(*int*)	auto	(*structure*)	palette

Defined in file:

C:/cal/Operations/SDI_Instruments/common/idl/core/xdibase__define.pro

XDIConsole

Inherits from: **XDIBASE**

Class Data:

(*structure*)	etalon	(*structure*)	camera	(*structure*)	header
(*structure*)	logging	(*structure*)	misc	(*structure*)	runtime
(*structure*)	buffer	(*string*)	obj_num	(*structure*)	geometry
(*int*)	need_frame	(*int*)	need_timer	(*int*)	auto
(*structure*)	palette	(*obj*)	manager	(*obj*)	console

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/xdiconsole__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

schedule=schedule: No Doc

mode=mode: No Doc

settings=settings: No Doc

start_line=start_line: No Doc

Example Call:

$$result = \textbf{XDIConsole->}\ \textbf{init}(schedule = schedule,\ mode = mode,\ settings = settings,\ start\_line = start\_line)$$

(pro) CAM_COOLER

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

$$\textbf{XDIConsole->}\ \textbf{cam\_cooler}, event$$

(pro) CAM_COOLER_EVENT

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> cam_cooler_event, *event*

(pro) CAM_EXPTIME

Method Documentation:
No Doc
Arguments:

event: No Doc

new_time=new_time: No Doc

Example Call:

XDIConsole–> cam_exptime, *event*,
new_time = new_time

(pro) CAM_GAIN

Method Documentation:
No Doc
Arguments:

event: No Doc

new_gain=new_gain: No Doc

Example Call:

XDIConsole–> cam_gain, *event*,
new_gain = new_gain

(pro) CAM_INITIALIZE

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> cam_initialize, *event*

(pro) CAM_SHUTDOWN

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> cam_shutdown, *event*

(pro) CAM_SHUTTERCLOSE

Method Documentation:
No Doc
Arguments:

event: No Doc

shutdown=shutdown: No Doc

Example Call:

XDIConsole–> cam_shutterclose, *event*,
shutdown = shutdown

(pro) CAM_SHUTTEROPEN

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> cam_shutteropen, *event*

(pro) CAM_STATUS

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> cam_status, *event*

(pro) CAM_TEMP

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> cam_temp, *event*

(pro) CLEANUP

Method Documentation:
No Doc
Takes no arguments
Example Call:

XDIConsole–> cleanup

(pro) CLOSE_MPORT

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> close_mport, *event*

(pro) EDIT_PORTS

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> edit_ports, *event*

(pro) EDIT_SETTINGS

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> edit_settings, *event*

(pro) EDITOR_CLOSED

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> editor_closed, *event*

(pro) END_AUTO_OBJECT

Method Documentation:
No Doc
Arguments:

id: No Doc

ref: No Doc

kill=kill: No Doc

Example Call:

XDIConsole–> end_auto_object, *id*,
ref,
kill = kill

(pro) EVENT_HANDLER

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

XDIConsole–> Event_Handler, *event*

(pro) EXECUTE_SCHEDULE

Method Documentation:
No Doc
Takes no arguments
Example Call:

XDIConsole–> execute_schedule

(pro) FILE_CHANGE_SCHED

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

XDIConsole–> file_change_sched, *event*

(pro) FILE_RE_INITIALIZE

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> file_re_initialize, *event*

(pro) FILE_SHOW

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> file_show, *event*

(pro) FILE_SHOW_SCHED

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> file_show_sched, *event*

(function) FORCE_IMAGE_UPDATE

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **XDIConsole–> force_image_update**()

(pro) GET_CAMERA_TEMP

Method Documentation:
No Doc
Arguments:

temp: No Doc

temp_state: No Doc

set_point: No Doc

Example Call:

XDIConsole–> get_camera_temp, *temp*,
temp_state,
set_point

(function) GET_DEFAULT_PATH

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **XDIConsole–> get_default_path**()

(function) GET_DLL_NAME

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **XDIConsole–> get_dll_name**()

(function) GET_ETALON_INFO

Method Documentation:
No Doc
Takes no arguments
Example Call:

result = **XDIConsole–> get_etalon_info**()

(function) GET_HEADER_INFO

Method Documentation:
No Doc

Takes no arguments
Example Call:

$$result = \textbf{XDIConsole}\text{–> }\textbf{get\_header\_info}()$$

(function) GET_IMAGE

Method Documentation:
No Doc
Arguments:

image: No Doc

Example Call:

$$result = \textbf{XDIConsole}\text{–> }\textbf{get\_image}(image)$$

(function) GET_LOGGING_INFO

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIConsole}\text{–> }\textbf{get\_logging\_info}()$$

(function) GET_PALETTE

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIConsole}\text{–> }\textbf{get\_palette}()$$

(function) GET_PHASE_MAP_PATH

Method Documentation:
No Doc
Takes no arguments

Example Call:

$$result = \textbf{XDIConsole->} \textbf{get\_phase\_map\_path}()$$

(pro) GET_PHASEMAP

Method Documentation:
No Doc
Arguments:

phasemap_base: No Doc

phasemap_grad: No Doc

phasemap_lambda: No Doc

Example Call:

$$\textbf{XDIConsole->} \textbf{get\_phasemap}, \textit{phasemap\_base}, \textit{phasemap\_grad}, \textit{phasemap\_lambda}$$

(function) GET_PORT_MAP

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIConsole->} \textbf{get\_port\_map}()$$

(function) GET_RAW_IMAGE

Method Documentation:
No Doc
Arguments:

image: No Doc

Example Call:

$$result = \textbf{XDIConsole->} \textbf{get\_raw\_image}(\textit{image})$$

(function) GET_SNR_PER_SCAN

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIConsole->}\ \textbf{get\_snr\_per\_scan}()$$

(pro) GET_SOURCE_MAP

Method Documentation:
No Doc
Arguments:

smap: No Doc

Example Call:

$$\textbf{XDIConsole->}\ \textbf{get\_source\_map}, smap$$

(function) GET_SPEC_SAVE_INFO

Method Documentation:
No Doc
Arguments:

nrings: No Doc

Example Call:

$$result = \textbf{XDIConsole->}\ \textbf{get\_spec\_save\_info}(nrings)$$

(function) GET_SPECTRA_PATH

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIConsole->}\ \textbf{get\_spectra\_path}()$$

(function) GET_TIME_NAME_FORMAT

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIConsole}\text{–> }\textbf{get\_time\_name\_format}()$$

(function) GET_ZONE_SET_PATH

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIConsole}\text{–> }\textbf{get\_zone\_set\_path}()$$

(pro) IMAGE_CAPTURE

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

$$\textbf{XDIConsole}\text{–> }\textbf{image\_capture}, \mathit{event}$$

(pro) KILL_HANDLER

Method Documentation:
No Doc
Arguments:

id: No Doc

kill_widget=kill_widget: No Doc

Example Call:

$$\textbf{XDIConsole}\text{–> }\textbf{Kill\_Handler}, \mathit{id}, \mathit{kill\_widget} = \mathit{kill\_widget}$$

(pro) LOAD_SETTINGS

Method Documentation:
No Doc
Arguments:

event: No Doc

filename=filename: No Doc

error=error: No Doc

first_call=first_call: No Doc

Example Call:

XDIConsole–> load_settings, *event*,
$filename = filename$,
$error = error$,
$first\_call = first\_call$

(pro) LOG

Method Documentation:
No Doc
Arguments:

entry: No Doc

sender: No Doc

display_entry=display_entry: No Doc

Example Call:

XDIConsole–> log, *entry*,
sender,
$display\_entry = display\_entry$

(pro) MODE_SWITCH

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> mode_switch, *event*

(pro) MOT_DRIVE_CAL

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> mot_drive_cal, *event*

(pro) MOT_DRIVE_SKY

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> mot_drive_sky, *event*

(pro) MOT_HOME_CAL

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> mot_home_cal, *event*

(pro) MOT_HOME_SKY

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> mot_home_sky, *event*

(pro) MOT_HOME_SOURCE

Method Documentation:
No Doc
Arguments:
image: No Doc

Example Call:

XDIConsole–> mot_home_source, *image*

(pro) MOT_SEL_CAL

Method Documentation:
No Doc
Arguments:
event: No Doc
set_source=set_source: No Doc

Example Call:

XDIConsole–> mot_sel_cal, *event*,
set_source = set_source

(pro) MOT_SEL_FILTER

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

XDIConsole–> mot_sel_filter, *event*

(pro) OPEN_MPORT

Method Documentation:
No Doc
Arguments:
event: No Doc

Example Call:

XDIConsole–> open_mport, *event*

(pro) REFRESH_SPEC_PMAPS

Method Documentation:
No Doc
Takes no arguments
Example Call:

XDIConsole–> refresh_spec_pmaps

(pro) SAVE_CURRENT_SETTINGS

Method Documentation:
No Doc
Arguments:

filename=filename: No Doc

Example Call:

XDIConsole–> save_current_settings, $filename = filename$

(pro) SCAN_ETALON

Method Documentation:
No Doc
Arguments:

caller: No Doc
start_scan=start_scan: No Doc
stop_scan=stop_scan: No Doc
pause_scan=pause_scan: No Doc
cont_scan=cont_scan: No Doc
start_volt_offset=start_volt_offset: No Doc
stop_volt_offset=stop_volt_offset: No Doc
volt_step_size=volt_step_size: No Doc
status=status: No Doc
reference=reference: No Doc
get_ref=get_ref: No Doc
wavelength=wavelength: No Doc
force_start=force_start: No Doc

Example Call:

XDIConsole–> scan_etalon, *caller*,
start_scan = *start_scan*,
stop_scan = *stop_scan*,
pause_scan = *pause_scan*,
cont_scan = *cont_scan*,
start_volt_offset = *start_volt_offset*,
stop_volt_offset = *stop_volt_offset*,
volt_step_size = *volt_step_size*,
status = *status*,
reference = *reference*,
get_ref = *get_ref*,
wavelength = *wavelength*,
force_start = *force_start*

(pro) SEE_CALIBRATION

Method Documentation:
No Doc
Arguments:

event: No Doc

Example Call:

XDIConsole–> see_calibration, *event*

(pro) SET_CENTER

Method Documentation:
No Doc
Arguments:

xcen: No Doc

ycen: No Doc

Example Call:

XDIConsole–> set_center, *xcen*,
ycen

(pro) SET_NM_PER_STEP

Method Documentation:
No Doc
Arguments:

nm_per_step: No Doc

Example Call:

XDIConsole–> set_nm_per_step, *nm_per_step*

(pro) SET_PHASEMAP

Method Documentation:
No Doc
Arguments:

phasemap_base: No Doc
phasemap_grad: No Doc
phasemap_lambda: No Doc

Example Call:

XDIConsole–> set_phasemap, *phasemap_base*,
phasemap_grad,
phasemap_lambda

(pro) SET_SNR_PER_SCAN

Method Documentation:
No Doc
Arguments:

snr: No Doc

Example Call:

XDIConsole–> set_snr_per_scan, *snr*

(pro) SET_SOURCE_MAP

Method Documentation:
No Doc
Arguments:

smap: No Doc

Example Call:

XDIConsole–> set_source_map, *smap*

(pro) SHUTDOWN_SPEX

Method Documentation:
No Doc
Takes no arguments
Example Call:

XDIConsole–> shutdown_spex

(pro) SPECTRUM_SNAPSHOT

Method Documentation:
No Doc
Arguments:

snapshot: No Doc

Example Call:

XDIConsole–> spectrum_snapshot, *snapshot*

(pro) START_PLUGIN

Method Documentation:
No Doc
Arguments:

event: No Doc

args=args: No Doc

new_obj=new_obj: No Doc

Example Call:

XDIConsole–> start_plugin, *event*,
$args = args$,
$new\_obj = new\_obj$

(pro) TIMER_EVENT

Method Documentation:
No Doc
Takes no arguments
Example Call:

XDIConsole–> timer_event

(pro) UPDATE_CAMERA

Method Documentation:
No Doc
Arguments:

commands: No Doc

results: No Doc

Example Call:

XDIConsole–> update_camera, *commands*,
results

(pro) UPDATE_LEGS

Method Documentation:
No Doc
Arguments:

leg1=leg1: No Doc

leg2=leg2: No Doc

leg3=leg3: No Doc

legs=legs: No Doc

Example Call:

XDIConsole–> update_legs, $leg1 = leg1$,
$leg2 = leg2$,
$leg3 = leg3$,
$legs = legs$

XDILog

Inherits from: **None**
Class Data:

(*string*)	log	(*long*)	log_window	(*string*)	prog_name
(*string*)	log_path	(*int*)	show_log	(*string*)	curdate

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/XDILog__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

log_window=log_window: No Doc
show_log=show_log: No Doc
prog_name=prog_name: No Doc
log_path=log_path: No Doc
log_append=log_append: No Doc
enabled=enabled: No Doc
header=header: No Doc

Example Call:

$$result = \textbf{XDILog}\text{->}\ \textbf{init}(log\_window = log\_window,\ show\_log = show\_log,\ prog\_name = prog\_name,\ log\_path = log\_path,\ log\_append = log\_append,\ enabled = enabled,\ header = header)$$

(pro) REFRESH

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$\textbf{XDILog}\text{->}\ \textbf{refresh}$$

(pro) UPDATE

Method Documentation:
No Doc
Arguments:

entry: No Doc

Example Call:

XDILog–> update, *entry*

XDIWidgetReg

Inherits from: **None**
Class Data:

(*long*)	id	(*string*)	type	(*obj*)	ref
(*int*)	store	(*int*)	need_timer	(*int*)	need_frame

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/xdiwidgetreg__define.pro

---

METHODS:

(function) INIT

Method Documentation:
No Doc
Arguments:

ref=ref: No Doc

id=id: No Doc

Example Call:

$$result = \textbf{XDIWidgetReg}\text{->}\ \textbf{init}(ref = ref, id = id)$$

(function) COUNT_OBJECTS

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIWidgetReg}\text{->}\ \textbf{count\_objects}()$$

(pro) DELETE_INSTANCE

Method Documentation:
No Doc
Arguments:

id: No Doc

Example Call:

$$\textbf{XDIWidgetReg}\text{->}\ \textbf{delete\_instance}, id$$

(function) GENERATE_LIST

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$result = \textbf{XDIWidgetReg}\text{–>}\ \textbf{generate\_list}()$$

(function) MATCH_REGISTER_FRAME

Method Documentation:
No Doc
Arguments:
id: No Doc

Example Call:

$$result = \textbf{XDIWidgetReg}\text{–>}\ \textbf{match\_register\_frame}(id)$$

(function) MATCH_REGISTER_FROM_TYPE

Method Documentation:
No Doc
Arguments:
type: No Doc

Example Call:

$$result = \textbf{XDIWidgetReg}\text{–>}\ \textbf{match\_register\_from\_type}(type)$$

(function) MATCH_REGISTER_REF

Method Documentation:
No Doc
Arguments:
id: No Doc

Example Call:

$$result = \textbf{XDIWidgetReg}\text{–>}\ \textbf{match\_register\_ref}(id)$$

(function) MATCH_REGISTER_STORE

Method Documentation:
No Doc
Arguments:

id: No Doc

Example Call:

$$result = \textbf{XDIWidgetReg}\text{–>}\ \textbf{match\_register\_store}(id)$$

(function) MATCH_REGISTER_TIMER

Method Documentation:
No Doc
Arguments:

id: No Doc

Example Call:

$$result = \textbf{XDIWidgetReg}\text{–>}\ \textbf{match\_register\_timer}(id)$$

(function) MATCH_REGISTER_TYPE

Method Documentation:
No Doc
Arguments:

id: No Doc

Example Call:

$$result = \textbf{XDIWidgetReg}\text{–>}\ \textbf{match\_register\_type}(id)$$

(pro) PRINT_REGISTER

Method Documentation:
No Doc
Takes no arguments
Example Call:

$$\textbf{XDIWidgetReg}\text{–>}\ \textbf{print\_register}$$

(pro) REGISTER

Method Documentation:
No Doc
Arguments:

id: No Doc
ref: No Doc
type: No Doc
store: No Doc
timer: No Doc
frame: No Doc

Example Call:

XDIWidgetReg–> register, *id*,
ref,
type,
store,
timer,
frame

(pro) SAVE_SETTINGS

Method Documentation:
No Doc
Arguments:

path: No Doc
id: No Doc
owner: No Doc
ref: No Doc

Example Call:

XDIWidgetReg–> save_settings, *path*,
id,
owner,
ref

(pro) SET_CONTROL

Method Documentation:
No Doc
Arguments:

id: No Doc

ref: No Doc

control: No Doc

Example Call:

XDIWidgetReg–> set_control, *id*,
ref,
control

Functions

(function) GET_ERROR

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/get_error.pro
Function Documentation:
Return an ANDOR error string given an error code.
Arguments:

err_code: Error code

(function) GET_NAMES

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/get_names.pro
Function Documentation:
From a full path list of plugins, return only the plugin names
Arguments:

path_list: Vector of plugin full path names

(function) ACE_FILTER_INTERFACE

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/ace_filter_interface.pro
Function Documentation:
Sends commands to an ACE filter wheel (used only at Poker I guess,
since com ports are hard coded here.
Arguments:

command=command: Command to send

(function) DRIVE_MOTOR

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/drive_motor.pro
Function Documentation:
Wrapper for controlling Fualhaber motors. Open/close ports, enable/disable motor,
get status, set position, drive to position, set speed/accel, drive in a direction
in small increments until blocked (i.e. when homing the mirror motor) etc.
Arguments:

port: Com port of the motor
dll_name: SDI_External dll name (full path)
direction=direction: String direction ("forwards" or "backwards") to drive until blocked
gohix=gohix: Drive to nearest hall index
goix=goix:
drive_to=drive_to: Drive to absolute position
control=control: String control command (see function body)
readpos=readpos: Read the motor position (returned from the function)
speed=speed: Set the speed
accel=accel: Set the acceleration
verbatim=verbatim: Send a string command verbatim to the motor, appending a carriage return
home_max_spin_time=home_max_spin_time: Max time to spin (for every small increment) when homing
timeout=timeout: Timeout in seconds

(function) GET_PATHS

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/get_paths.pro
Function Documentation:
No Doc
Takes no arguments

(function) GET_SUN_ELEVATION

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/get_sun_elevation.pro
Function Documentation:
Get the current sun elevation for a given latitude and longitude.
Arguments:

lat: Geographic latitude
lon: Geographic longitude

(function) PHASEMAP_UNWRAP

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/phasemap_unwrap.pro
Function Documentation:
'Unwrap' a phasemap produced by the SDIPhasemapper plugin.
Arguments:

xcen: Nominal x center

ycen: Nominal y center

radial_chunk: Size of the chunk over which to average the phase (value of 50 is used in phasemapper)

channels: Number of channels in the scan

threshold: Value of 80 is used by the phasemapper

wavelength: The wavelength at which the phasemap was recorded

phasemap: The actual phasemap 2D array

show=show: Show the unwrap as it occurs

tv_id=tv_id: Id of the tv window for showing the unwrap

dims=dims: Dimensions of the tv window for drawing

(function) ZONEMAPPER

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/zonemapper.pro
Function Documentation:
Function/method/pro documentation here
Arguments:

nx: Arg0

ny: Arg1

cent: Arg2

rads: Arg3

secs: Arg4

nums: Arg5

show=show: Arg6

outang=outang: No Arg Doc

outrad=outrad: No Arg Doc

Procedures

(pro) GET_EPHEMERIS

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/get_ephemeris.pro
Procedure Documentation:
No Doc
Arguments:

save_name=save_name: No Doc

safe_sea=safe_sea: No Doc

lat=lat: No Doc

lon=lon: No Doc

timeres=timeres: No Doc

start_stop_times=start_stop_times: No Doc

get_sea=get_sea: No Doc

(pro) HANDLE_ERROR

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/SDI_Main.pro
Procedure Documentation:
Error handler.
Arguments:

error: Error recieved

(pro) HANDLE_EVENT

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/SDI_Main.pro
Procedure Documentation:
Handle widget events. These are rerouted to the console's event handler.
Arguments:

event: Widget event structure

(pro) KILL_ENTRY

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/SDI_Main.pro
Procedure Documentation:
Handle widget destroy events. These are rerouted to the consoles kill handler.
Arguments:

id: Widget id

(pro) MARKS_PALETTE

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/load_pal.pro
Procedure Documentation:
No Doc
Takes no arguments

(pro) SDI_MAIN

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/SDI_Main.pro
Procedure Documentation:
SDI entry point, called with a settings file, optional schedule and optional mode.
Arguments:

settings=settings: Settings file (required)

schedule=schedule: Schedule file (required if mode is "auto")

mode=mode: String mode, "auto" or "manual", defaults to "manual"

(pro) TREE_CLEANUP

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/edit_console_settings.pro
Procedure Documentation:
If this editor was created by the SDI console, alert it that we have closed.
Arguments:

id: Widget id

(pro) TREE_EVENT

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/edit_console_settings.pro
Procedure Documentation:
Handle events generated by the tree widget.
Arguments:

event: Widget event structure

(pro) WRITE_SPECTRA_NETCDF

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/write_spectra_netcdf.pro
Procedure Documentation:
No Doc
Arguments:

ncdid: No Doc

spectra: No Doc

start_time: No Doc

end_time: No Doc

nscans: No Doc

acc_im: No Doc

create=create: No Doc

fname=fname: No Doc

return_id=return_id: No Doc

header=header: No Doc

data=data: No Doc

reopen=reopen: No Doc

update=update: No Doc

(pro) COMMS_WRAPPER

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/comms_wrapper.pro
Procedure Documentation:
No Doc
Arguments:
port: No Doc
dll_name: No Doc
type=type: No Doc
: No Doc

(pro) CONSOLE_CRASH_ROUTINE

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/crash_routines.pro
Procedure Documentation:
Check to see if the console 'crash' file is present. If it is, it is likely that the
SDI console has stopped running, and this gets logged.
Arguments:
log_file: The filename to send/append log output to

(pro) CONSOLE_MAKE_CRASH_FILE

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/crash_routines.pro
Procedure Documentation:
Create the console 'crash' file.
Arguments:
crash_file: Filename for the crash file

(pro) CRASH_ROUTINES

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/crash_routines.pro
Procedure Documentation:
This gets called by a Windows scheduled script, and checks to see if a crash file is
present (the console should delete this file, so if it is present, the console has likely
crashed), and if so it logs a crash. If not ,it recreates the file.
Takes no arguments

(pro) DEFINE_VARIABLES

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/edit_console_settings.pro
Procedure Documentation:
Create the SDI variables/structures.
Arguments:

var_holder: Variables will be returned in this structure

(pro) DRIVE_MOTOR_WAIT_FOR_POSITION

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/drive_motor.pro
Procedure Documentation:
Wait for a position reached notification from the motor (a 'p' character). A timeout can be provided to prevent waiting forever.
Arguments:

port: Com port for the motor

dll_name: Name of the SDI_External dll

com: String 'com' type, e.g. "moxa"

max_wait_time=max_wait_time: Max time to wait in seconds

errcode=errcode: Returned error code

(pro) EDIT_CONSOLE_SETTINGS

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/edit_console_settings.pro
Procedure Documentation:
Entry point for the console settings editor. Can be called ddirectly from IDL command line, or from the SDI console.
Arguments:

filename=filename: Pass in a filename to load upon startup

leader=leader: Widget leader, when called from the console

console=console: The console object reference, if started from the console

---

(pro) EDIT_LOAD_SETTINGS

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/edit_console_settings.pro
Procedure Documentation:
Load a settings file from disk.
Arguments:

filename=filename: Filename to load

(pro) EDIT_PORT_SETTINGS

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/edit_console_settings.pro
Procedure Documentation:
Create an xvaredit dialog for editing the port structure.
Takes no arguments

(pro) EDIT_SAVE_SETTINGS

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/edit_console_settings.pro
Procedure Documentation:
Save the current settings.
Arguments:

filename=filename: Filename to save to

nosplash=nosplash: Optionally hide the "File saved" dialog

(pro) GET_JD0_SEC

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/get_jd0_sec.pro
Procedure Documentation:
Get the current julian date and the seconds into the day.
Arguments:

jd0: OUT: Julian date at midnight I think...

sec: OUT: Seconds into the julian day

(pro) LOAD_PAL

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/load_pal.pro
Procedure Documentation:
No Doc
Arguments:

culz: No Doc

idl_table=itbl: No Doc

bright=brt: No Doc

proportion=prp: No Doc

(pro) PAL_SUBSAMP

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/load_pal.pro
Procedure Documentation:
No Doc
Arguments:

idxlo: No Doc

idxhi: No Doc

sred: No Doc

sgrn: No Doc

sblu: No Doc

brt: No Doc

satval: No Doc

sign: No Doc

(pro) RESTART_MOXA

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/restart_moxa.pro
Procedure Documentation:
Restart the MOXA USB hub, using pstools (TODO: is this used? Paths are hard coded...)
Takes no arguments

(pro) SCHEDULE_READER

Defined in file:
C:/cal/Operations/SDI_Instruments/common/idl/core/schedule_reader.pro
Procedure Documentation:
Query an SDI schedule file for the next command.
Arguments:

schedule_file: Schedule file name

schedule_line: The current schedule line

xcomm: OUT: string command

xargs: OUT: string array of arguments

lat: Geographic latitude

lon: Geographic longitude

console_ref: Object reference for the console

refresh_nm_per_step=refresh_nm_per_step: Look for a nm per step refresh command (special syntax)

refresh_phasemap=refresh_phasemap: Look for a phasemap refresh command (special syntax)
