DISR Data Software Companion Document
Version 1.0
28 September 2012

Contents

Introduction...

The Descent Image & Spectral Radiometer (DISR) data from the Huygens probe is generally analyzed using 3 tools:

1) Research Systems Incorporated (RSI)'s Interactive Data Language (IDL),
2) Microsoft Excel Spreadsheets, and
3) Fortran

A collection of data manipulation routines has been developed in IDL for the DISR instrument, named DISRSOFT. It allows easy access to the data. In DISRSOFT there are commands to read the header and pixel data into IDL arrays, which can then be printed into tables or displayed as 2D plots, 3D surfaces, contours, or images by investigators using other, ad hoc, IDL routines. Once extracted from the datasets analysis is often performed using Excel spreadsheets. Examples of these spreadsheets exist in many of the IDL_PROGRAMS subdirectories. Fortran was also occasionally used to reduce the DISR data, most notably for the radiative transfer models and descent simulations.

The DISR flight data volume is discussed in the document: "DISR XDR Data Archive Document.doc", which is in the Planetary Data System (PDS) archive. In summary, the data is collected and organized by the instrument's Ground Support Equipment (GSE) into External Data Representation (XDR) format datasets (generally one dataset per observation) which are stored in a particular directory structure. This document discusses the software used to access and analyze that data.

DISRSOFT...

DISRSOFT is a collection of IDL routines, developed by the project, to allow easy extraction, manipulation, and display of the DISR's XDR datasets. A text document titled "DISRSOFT_DOC.rtf" exists in the IDL_PROGRAMS\OTHER\CAL_SOFT directory which describes the DISRSOFT concept and many of the associated IDL routines. The document was initially very optimistic, purporting to "..describe everything necessary for the calibration of DISR...", however it does not. Also, it was expanded to incorporate similar software for the Imager for Mars Pathfinder (IMP). References to the IMP software can be ignored by the DISR user. DISRSOFT_DOC does have a good description of the DISR header information, and the primary routines necessary to access the data. It also describes how to 'install' the DISRSOFT routines into the IDL directory structure.

In general 'installing' DISRSOFT involves placing the routines in a known directory, and then directing IDL to that directory. Here are the basic steps...

1) Copy the CAL_SOFT directory from \IDL_PROGRAMS\OTHER to the root directory of your machine (typically C:\).

2) Using a text editor, modify the idl_start.pro routine, found in the CAL_SOFT directory to add the location of your IDL routine depository to the path thustly:
!path = !path + {your idl repository directory}
As an example I typically use: !path=!path+'C:\idl_programs\csee'
Note the added path must be in quotes.

3) In UNIX it is typically necessary to run the start program from the IDL prompt with the batch command @C:/CAL_SOFT/idl_start.pro.
In Windows, one can have the start program to automatically execute by adding it to the IDL preferences:
File -> Preferences -> Startup:
IDL Main directory = C:\RSI\IDL
Working Directory = C:\
Satrtup file = C:\CAL_SOFT\idl_start.pro

Hardware Platform...

The software developed during the DISR program on three different platforms: HP Unix, Apple Macintosh and Windows PC. Although many IDL routines transfer well from one platform to the other, no serious effort was made to keep these programs platform independent. It may be necessary to change some lines of code to accommodate a platform other than the one the routine was developed on.

Contents of the IDL_PROGRAMS directory...

The IDL_PROGRAMS directory is generally a collection of the IDL programs used for the analysis of the DISR data, organized by author. There is no further sub-organization. The IDL_PROGRAMS\OTHER subdirectory contains the DISRSOFT code (CAL_SOFT), the flat field values used in the DISR's on-board software, and the IDL code for the Windows based GSE program.

Flat Fields...

The \IDL_PROGRAMS\OTHER\FLAT_FIELD directory contains the flat field information for each DISR imager. The flight unit, which landed on Titan is DISR #3. The files present the 'correction_factor', which is to be divided into the image, as described in 5.8 of the DISR Users' Guide. This information is presented in 4 formats:

1) The files with no extension are the raw correction_factor values in tabular form (corresponding to row and column of the specific imager).

2) The files with the ".txt" extension contain the same data plus imager identification information, and row & column headings.

3) The ".csv" files contain the same information in Comma Separated Variable format.

4) The ".disrsoft" files contain the same data plus the header information contained in the "...Header..." files in XDR format.

NEW_GSE...

The 'new' gse program is the Windows based Ground Support Equipment data stream decoding software, which runs under IDL. It takes the European Space Operations Center (ESOC) supplied data stream (i.e. file o500sd_.1h_), and produces the XDR formatted DISR datasets. In order for the batch program (gse) to work correctly the files from the \IDL_PROGRAMS\OTHER\NEW_GSE directory must be copied to 'C:\idl_programs\new_gse (or modify the batch program). Once this is done, one can follow the directions in "\IDL_PROGRAMS\OTHER\NEW_GSE\Instructions for new3.doc" to run the program.

Frequently used commands, routine or functions...

Here are some of the most commonly used DISRSOFT commands:

d_read,*dataset_name*,h,p ;reads the header and pixel information from the XDR formatted dataset (*dataset_name*) into arrays h & p respectively.

d_irread,*dataset_name*,h,p,g ;reads the header, pixel & bin information from the XDR formatted IR spectrometer dataset (*dataset_name*) into arrays h, p & g respectively.

d_whelp ; help GUI

d_look ; image display GUI

Exposure=d_value(h,36) ; reads the value for the exposure time from the header structure, and places it in variable 'Exposure'. d_value is also applicable to all other header entries:

d_value		Name	Description
0		H_TYPE	Header type (D=DISR, I=IMP)
1		H_DIRECTORY	Directory file last stored in
2		H_FILENAME	Filename file was stored as
3		H_FILETIME	Time file was last written by D_WRITE
4		H_LENGTH	Length of header string array
5		H_DIMENSION	Number of dimensions in data
6		H_XSIZE	Number of columns
7		H_YSIZE	Number of rows
8		H_ZSIZE	Number of images
9		H_DATATYPE	Type of data
13		H_PURPOSE	Purpose of observation
14		H_SUBJECT	Subject of observation
15		H_DATE	Date observation made
16		H_ENGINEER	Engineer responsible
17		H_SITE	Site of observation
18		H_SET_NAME	Text description of data set
22		H_CCDTEMP	On chip CCD temperature in [K]
23		H_REF_TEMP	CCD electronics reference temperature (MPAE)
30		H_DETECTOR	detector type (CCD, IR, PHOTOMETER)
31		H_SENSOR_ID	sensor id number
36		H_EXPTIME	integration time [ms] for CCD measurement
37		H_IMAGE_TIME	start time of exposure
50		H_COORD_XL	image coordinates (lower x)
51		H_COORD_YL	image coordinates (lower y)
54		H_FILTER	filter in use
62		H_VACUUM	vacuum or not (yes or no)
63		H_OPTICS	additional optics on bench (MPAE)
64		H_TESTLAMP	external test lamp (MPAE)
80		H_GSE_REV	GSE revision creating data set

d_value		Name	Description
81		H_CONFIG_FILE	configuration file associated with data set
82		H_TEST_LOG	test log filename generating data
83		H_REC_BEG	first record of this data set in test log
84		H_REC_END	last record of this data set in test log
85		H_PIXEL_UNITS	units of pixels
86		H_SET_ID	data set id
87		H_SET_NUMBER	sequential number of this data set
88		H_MISSION_TIME	mission time in seconds
89		H_CYCLE_NUMBER	sequential numbering of cycles
90		H_CYCLE_TYPE	cycle type
91		H_SCEN_STEP	cycle criteria table entry number
92		H_SPM_FLAG	spectrophotometric flag
93		H_CCD_MEAS	number of CCD sets performed during cycle
94		H_IR_MEAS	number of IR sets performed during cycle
95		H_VIOLET_MEAS	number of violet sets performed during cycle
96		H_MEAS_TYPE	measurement type
97		H_AZIMUTH	azimuth at start of cycle
98		H_ALTITUDE	altitude at start of cycle
99		H_SPIN	spin rate at start of cycle
100		H_TARGET_AZ	target azimuth for measurement
101		H_ACTUAL_AZ	actual azimuth for measurement
102		H_LAMP_STATE	cal and surf lamp status
103		H_IR_STATUS	IR hardware status word
104		H_IR_FLAGS	IR sampling and compression flags
105		H_IR_CHP_TMPB	IR chip temperature (start)
106		H_IR_CHP_TMPE	IR chip temperature (end)
107		H_PRECHARGE	Average precharge voltage
108		H_IR_COL_TIME	IR collection time [ms]
109		H_NUM_ROT	number of region/rotations (IR)
110		H_NUM_REGIONS	number of regions (IR)
111		H_DC_OFFSET_U	DC Offset voltage for ULIS
112		H_DC_OFFSET_D	DC Offset voltage for DLIS
113		H_UIR_TGT_PRC	target %age for ULIS
114		H_DIR_TGT_PRC	target %age for DLIS
115		H_UIR_PRCTILE	%ile point used for ULIS
116		H_DIR_PRCTILE	%ile point used for DLIS
117		H_CCD_STATUS	CCD hardware status
118		H_DCS_STATUS	DCS hardware status
119		H_CCD_FLAGS	CCD flags
120		H_PROC_FLAGS	CCD processing flags
121		H_NUM_COL	number of columns transmitted
122		H_NULL_PIXEL2	sum of null pixels (col 2)
123		H_NULL_PIXEL3	sum of null pixels (col 3)
124		H_CCD_TGT_PRC	target %age for CCD
125		H_CCD_PRCTILE	%ile point for CCD
126		H_STP_CNT_COL	strip centre column
127		H_STP_FST_COL	1st column of strip

d_value		Name	Description
128		H_CAL_TMP_SORS	cal. source voltage
129		H_CCDLUG_TEMP	CCD thermal lug temperature [K}
130		H_STRAP_TEMP	thermal strap temperature
131		H_OPTICS_TEMP	conduit housing temperature
132		H_VIOLET_TEMP	violet detector temperature
133		H_SH_AUX_TEMP	sensor header aux board temperature
134		H_SH_BOX_TEMP	sensor header box temperature
135		H_EA_BOX_TEMP	EA box temperature
136		H_AUX_BRD_VOLT	divided 12V in volts
137		H_CPU_BRD_VOLT	divided 5V in volts
138		H_ADC_OFFSET	0V input; ADC offset
139		H_ADC_GAIN	2.5 input; gain of ADC
140		H_DISPQ_SIZE	max. dispatcher queue size
141		H_ALRMQ_SIZE	max. alarm queue size
142		H_TLMQ_SIZE	max. TM queue size
143		H_SCIPRQ_SIZE	max science processing size
144		H_STACK_SIZE	max stack size
145		H_CAL_SCENARIO	cal. scenario number
146		H_CCD_REP	CCD cal. repetition number
147		H_IR_REP	IR cal. repetition number
148		H_VIOLET_REP	violet cal. repetition number
149		H_SHUTTER_REP	shutter test rep. number
150		H_SHUTTER_CYC	number of shutter cycles
151		H_DCS_REP	DCS cal. repetition number
152		H_COMP_RATIO	compression ratio
153		H_HEATER_REP	heater test rep. number
154		H_HEATERS	which heaters
155		H_CAL_LAMP_REP	cal. lamp repetition number
156		H_NUM_BINS	number of IR bins for data collection
157		H_SURF_LAMP_REP	surface lamp rep. number
158		H_SUN_LAMP_REP	sun sensor LED rep. number
159		H_CAL1_VOLT1	voltage on cal. lamp 1
160		H_CAL1_CURR	current on cal. lamp 1
161		H_CAL2_VOLT1	voltage on cal. lamp 2
162		H_CAL2_CURR	current on cal. lamp 2
163		H_CAL3_VOLT1	voltage on cal. lamp 3
164		H_CAL3_CURR	current on cal. lamp 3
165		H_SURF_VOLT1	voltage on surface lamp
166		H_SURF_CURR	current on surface lamp
167		H_DCS_TST_STAT	overall DCS test status
168		H_DCS_SLF_STAT	DCS self test status
169		H_DCS_SW_STAT	DCS s/w test status
170		H_SUN_LED_VLT1	voltage on Sun sensor LED
171		H_SUN_LED_CURR	current on Sun sensor LED
172		H_SUN_RESP	Sun sensor reading with LED on
173		H_NUM_HEAT	number of heaters tested
174		H_NUM_TRIPLET	number of sun triplets in data set

d_value		Name	Description
175		H_NUM_TIME_PAIRS	number of time data pairs
176		H_DUMP_START	address of first word in dump
177		H_DUMP_LEN	number of words in dump set
178		H_DUMP_FLAG	packing flag
179		H_MESSAGE	message type code
180		H_MESSAGE_ID	additional information code
181		H_NUM_BAD	number of bad ranges in RAM or EEPROM
182		H_LOTS_BAD	flag - lots bad RAM or EEPROM areas
217		H_OBJECTIVE	Objective of test
218		H_DUMMY	Flag for dummy detector
219		H_MODEL	DISR model for MMC
222		H_SQRT_MIN	Minimum value for sqrt table
223		H_SQRT_MAX	Maximum value for sqrt table
225		H_SUN_SOURCE	source of spin data
226		H_OGSE_STATUS	status of OGSE
227		H_OGSE_MSG	last msg from EGSE displayed on OGSE
228		H_OGSE_LAMP	lamp current
229		H_OGSE_SUNFILE1	path of sun pulse file
230		H_OGSE_SUNFILE2	path of sun pulse file
231		H_OGSE_SAMP	sun amplitude current
232		H_OGSE_SUNTABLE	sun table
233		H_OGSE_RPM	sun pulse RPM
234		H_OGSE_SPAN	sun pulse span value
235		H_OGSE_SIDEC	Si detector reference
236		H_OGSE_GEDEC	Ge detector reference
237		H_OGSE_EXTDEC	external detector reference
238		H_OGSE_5V	5V reference
239		H_OGSE_TEMP	internal temperature
240		H_OGSE_SSS_TEMP	sun sensor stimulator temp.
241		H_OGSE_GN	analog ground noise
255		H_MONO_WAVL	Monochromator wavelength [nm]
256		H_MONO_SI	Monochromator Si detector reading
257		H_MONO_GE	Monochromator Ge detector reading
258		H_MONO_FLTR	Monochromator filter
259		H_MONO_G	Monochromator gain
260		H_MONO_RT	Monochromator response time [s]
261		H_MONO_GRAT	Monochromator grating
262		H_MONO_TUR	Monochromator turret number
263		H_SPSI	Sphere Si detector reading
264		H_SPFLSI	Sphere filtered Si reading
265		H_SPINGAS	Sphere InGaAs detector reading
266		H_SP_DETTMP	Sphere detector temperature
267		H_COSI	Collimator silicon detector reading
268		H_COFLSI	Collimator filtered Si reading
269		H_COINGAS	Collimator InGaAs detector reading
270		H_COTMP	Collimator detector temperature
271		H_PLMINGAS	Palmer InGaAs detector reading

d_value		Name	Description
272		H_PLMTMP	Palmer detector temperature
278		H_CPGASTEMP	Temperature of SH internal purge gas
279		H_AMBTEMP	Ambient temperature of clean room
280		H_SPTEMP	Temp[erature of gas in sphere
281		H_BOXTEMP	Temperature of gas in dry box
282		H_CHMBTEMP	Temperature in environment chamber
286		H_CAL_EL	Sensor head elevation
287		H_CAL_AZ	Sensor head azimuth
288		H_CAL_POL	Collimated beam polarizing angle
289		H_CAL_SHUT	Calibrator high intensity shutter state
290		H_LAST_CALTIME	Time of last cal. log entry

A more complete listing of the DISR header entries exists in the archive under:
\IDL_PROGRAMS\OTHER\CAL_SOFT\Headers2.xls

Some commonly used IDL utilities:

average = mean(*array*) ; returns the average value of an array

short_string = sstr(*long_string*) ; strips leading and trailing spaces from a string

time=hmsplus3(*seconds*) ; converts seconds to string of Hours:Min:Sec.XXXX

nsecs=seconds(*hms*) ; converts string in the form Hrs:Min:Sec to decimal seconds

path : prints out the directories in the current path

dir ; prints the current directory contents

up ; moves the current directory up one level

array_sort,*array* ; rearranges *array* into ascending order.

filtzero,*a*,*b*,n ; removes the common 0 elements of *a* & *b* and returns resulting size, n.

thumbnails2,*log* ; creates postscript thumbnails of the images in *log* (from DB2).

SingleImages3,*log* ; creates jpg, png & hist. equalized files of images in *log*.

@reset ; closes all files and resets parameters, echoing current directory.

PTEST...

PTEST is short for Probe Testing. The ptest, ptestUNIX and ptestPC batch routines were designed to get rapid instrument status and health during the cruise phase testing. These routines are hardware platform specific, with ptest and ptestUNIX being designed for HPUX machines and ptestPC (most recent) modified to work on Windows platforms. The ptest programs are interactive and call the following other IDL routines:

Routine	Output
thumbnails.pro	Prints thumbnail pictures of all the images. (.\PS\thumbnails.ps)
tripletfiles.pro	Creates jpeg, png and histogram equ. files of the triplets (.\triplets)
SingleImages.pro	Creates jpeg, png and histogram equ. files of the images (.\Images)
plot_temps.pro	Creates the time-ordered lists (voltages, temps, etc) & plots temperatures (plot_temps.out)
sun_sensor.pro	Tabularizes the pulse time, amplitude & spin rate for each rotation. (.post\anal_results\sun_pulses)
idx_check.pro	Time of first & last packets, Test duration, # of packets. (idx_check.out)
telem_packet_checker.pro	Reports time & # of lost packets. (telem_packet_ck.out)
plot_rate.pro	Plots the telemetry rate, altitude & spin vs. mission time. (telem_alt_spin_rate.out)
check_times.pro	Checks for missing or erratic DDB times. (.\time_checker.out)
cycles.pro	Summarizes the # of each type of cycle & lists spin and altitude. (cycles.out)
cycle_type_vs_alt_plot.pro	Plots & tabularizes the cycle progression vs. altitude. (cycle_types.out)
plot_spin.pro	Plots & tabularizes the spin rate vs. mission time. (spin_profile.out)
plot_alt.out	Plots & tabularizes the altitude vs. mission time. (alt_profile.out)
ds_summary3.pro	Creates a summary of the contents of the datasets. (DS_Summary)
time_checker.pro	Creates a report of time discrepancies. (time_checker.out)
count_DB_files.pro	Lists the number of each type of Data Base files. (DB_file_count.out)
resource.pro	Tallies the usage of the limited life items. (lamps & shutter) (resource.dat)
ccdstat.pro	Examines the CCD error flags for every CCD measurement.
lamplist.pro	Summarized the time and duration of lamp activations. (lamplist3.out)
lampdetail.pro	Lists the voltage and current for every lamp activation. (lamp_detail.out)
plot_drift.pro	Plots the clock drift vs. mission time. (clock_drift.out)
plot_time.pro	Plots the progression of the DISR and DDB time. (clock_time.out)
plot_tolist.pro	Plots all the parameters (voltages, temperatures, etc) from the TOL's.
sun_sensor_history	Plots the progression of the Sun Sensor amplitude vs time.
sun_lamp_history	Plots the Sun Sensor Stimulator LED power & associated counts (sun_lamp_history.out)

PTEST is run from the test log level (i.e. in the directory containing DB and/or DB2). The output from these routines are typically deposited into the test log under examination, and subsequently copied into the .\post\anal_results directory (in new directory: 'ptest'). One particularly valuable output file is DS_Summary, which is a chronological ASCII text listing of all the datasets collected.

Below are presented two example programs used to manipulate DISR data. The first interactively displays the DISR images in the data log, and the second prints out an ASCII list summarizing the IR spectrometer datasets.

IR spectrometer g structure...

The following is a rudimentary explanation of the IR spectrometer's 'g' data structure:

Besides having a header, the IR spectrometer data also contains a data structure with information on the azimuth at which the data was taken. There are 3 fundamental members to the IR g structure:

1) "Regions" or Regions Table (in the archive) defines the shape of the data collection bins (regions) in azimuth space and which bin (region) has ULVS or DLVS data. (member g.regions)
2) "Reading" or Reading Table chronologically lists the IR data accretion by rotation number and region (bin); giving mission time, Collection Time, Shutter Time & Sample Time for each accumulation. (member g.reading)
3) "Bins" or Bins Table relates the datasets pixel information (in the data table) to the bins (regions). For each column (in the archive data table), Bins tell whether the data is ULIS, DLIS, shutter open or shutter closed, plus the total number of samples & collection time for each bin. (member g.bins)

"Regions" is a 5 by N array where N is the number of regions each rotation is divided into (typically 1 or 8). Each line contains the following 5 elements:

(0,x) = Region number. Usually equal to x+1 (where x goes from 0 to N-1)

(1,x) = Start azimuth in 100th's of degrees.* Az_begin = g.regions(1,x)/100.0

(2,x) = Ending azimuth in 100th's of degrees.* Az_end = g.regions(2,x)/100.0

(3,x) = Up bin index. Identifies bin that ULIS data will be summed into

(4,x) = Down bin index. Identifies bin that DLIS data will be summed into

In the archive data the Regions table is arrange in 5 columns by N rows. Typically the DLVS is collected in 8 regions and the ULVS is collected in 4 regions.

"Reading" is a 6 by N array, where N is the product of the number of rotations completed and the number of regions (bins) used per rotation (i.e. 8 bins & 2 rotations yields 16 elements).

Each line contains the following:

(0,x) = Rotation number (1-N)

(1,x) = Region of rotation used. This often does NOT start with 1.

(2,x) = Mission time at start of the rotation/region read in 10,000th's of a second.
Mission time (sec) = g.reading(1,x)/10000.0

(3,x) = IR duration (collection time) in this rotation/region.
Duration (msec) = g.reading(3,x)*8.064

(4,x) = IR shutter period for this rotation/region. This is composed of period/2 closed, followed by period open, followed by period/2 closed in each shutter cycle. There is also a 16.128 msec shutter transition wait in between each open/close and close/open change.

(5,x) = IR sample time. This is the time to use to normalize all IR data into counts per second (equivalent to exposure time). Sample time (msec) = g.reading(5,x)*8.064

The shutter period should be an integer multiple of the sample time to assure proper IR operation.

In reading, the transform between duration (collection time), period (shutter time), and cycles is: :

Cycles = (duration(raw) - 2)/(2 * period(raw) + 4) = (g.reading(3,x) - 2) / (2 * g.reading(4,x) + 4)

Duration (msecs) = ((period(raw)*2*cycles + 4*cycles + 2)*8.064 msec

where (raw) is the number of 8.064 ms steps from the g.reading table, and cycles is the number or repetitions of the basic collection cycle (closed, open, closed).

The number of shutter cycles in a collection is not stored in the reading array, and must be deduced from the above equation.

"Bins", is a 6 by N array, where N in the number of columns of data in the pixel (data) array. It is not unusual for there to be a gap between the highest down bin number and the lowest up bin number (g.bins(0,*)).

Each line contains the follow entries:

(0,x) = Bin number. This is NOT a continuous sequence.

(1,x) = DLIS or ULIS bin. 0 = DLIS (down bin), 1 = ULIS (up bin)

(2,x) = IR shutter open or closed. 1 = closed, 0 = open

(3,x) = Total shutter open integration time over all rotations for that bin (region) in 10,000ths of a second. Total time (sec) = Bins (3,i)/10000.0. Since the datacontained in Bins (5,i) are four times the average counts in all the reads producedby the French digital output, not the total counts divided by the total integrationtime, do NOT divide the data numbers by Bins (3,i) to determine counts persecond. Counts/sec is obtained by dividing the data by the average sample timedefined below.

(4,x) = Total number of samples taken with shutter open for that bin (region)
g.bins(4,x) = summ(all rotations) of period*cycles/sample

(5,x) = Column number of the pixel array that corresponds to this bin.
p(*,g.bins(5,x) contains the averaged data for bin g.bins(0,x)

* Azimuths are defined as CCW, viewed from above, relative to the sun.

Note that the data recorded for each pixel in each region with the shutter closed or open is the average data number. The average sample time is the total collection time divided by the total number of samples taken on all rotations for that region, or Bins (3,i)/Bins (4,i). Thus the net count rate, DN/sec is:

$$DN/sec = (\text{data value closed} - \text{data value open}) / [Bins(3,i)/Bins(4,i)]$$

Example Program #1: DISR Image Display

```
pro images_ex,log

; CSee, 2012.07.16
; This program displays the images in data directory: log.
; It is to be run from the Log level (i.e. the directory containing DB).

db='DB2'    ;DB for UNIX data or DB2 for Windows data

print,''
print,'This program prints images frames from the specified log to the screen.'
print,'The default is the current test log.'


cd,'',current=cdir & cd,cdir        ;the current directory is the default log
if n_params() eq 0 then log = cdir+'\'  ;test for # of input parameters
if strmid(log,0,/reverse_offset) ne '\' then log=log+'\'    ;add final '\' if missing

print,''
print,'The operant Test Log is: ',log

;____________________________________________________________________________
;List Files...

cd,log+db+'\Image\',current=cdir
files=file_search()
cd,cdir

s=size(files)
nfiles=fix(s(1))
numfiles=sstr(nfiles)         ;numfiles=short string (nfiles)

if files(0) eq '' then begin & print,'No files found' & goto,exit & endif

print,'The number of Images in this log is: ',sstr(nfiles)

;____________________________________________________________________________
;Initialize...
```

```
auto=1    ;for auto=1 tvscl
min=0     ;the tvscl min setting
max=4095  ;the tvscl max setting
ilast=0   ;ilast is the previous value of i
ff=-1     ;flag for flat fields, 1=>enabled, -1=>disabled

menu:

print,''
print,'Enter  to...              Enter  to...
print,'  0    Get this menu       -4    Autoscale, min to max
print,' -1    Quit the program    -5    Scale from min to 2 x average
print,' -2    Change minimum      -6    Print current image
print,' -3    Change maximum      -7    Toggle flat field   -8  Jpeg image.

;____________________________________________________________________________
;Loop that reads the desires...

again:
 i=0    ;make i an integer
 print,'Pick a file (from 1 to ',sstr(nfiles),') or choose menu (0): '
 read,i               ;i=file index

if i eq 0 then goto,menu    ;display menu

if i eq -1 then goto, exit  ;end of program

 if i eq -2 then begin      ;set minimum of stretch
  auto=0
  print,'Min = ', sstr(min)
  read,'Enter new min: ',min
  i=ilast
 endif

 if i eq -3 then begin      ;set maximum of stretch
  auto=0
  print,'Max = ',sstr(max)
  read,'Enter new max: ',max
  i=ilast
 endif
```

```
if i eq -4 then begin       ;autoscale to max and min
 auto=1
 i=ilast
endif

if i eq -5 then begin        ;autoscale, range = minimum to 2 x average
 auto=2
 i=ilast
endif

if i eq -6 then begin   ;prints current image
 set_plot,'ps'
 device,filename = 'c:\idl.ps',/times,/portrait,xsize=7.5,ysize=9,yoffset=1.0,/inches,bits_per_pixel=8
 if auto eq 1 then tvscl,bytscl(p,min=min(p),max=max(p),top=255) else $
  tvscl,bytscl(p,min=min,max=max,top=255)  ;p>min<max
 xyouts,0,-.02,typ+sstr(ilast)+'  Min= '+sstr(min(p))+', Ave= '+sstr(mean(p))+', Max= '+sstr(max(p)),/norm
 xyouts,0,-.04,'Exposure (ms) = '+sstr(d_value(h,36))+'   lamps = '+sstr(d_value(h,102)),/norm
 xyouts,0,1.02,'From Log: '+cdir,/norm
 device,/close_file
 spawn,'gsview32 c:\idl.ps'  ;requires ghostscript viewer to print.
 ;set_plot,'x'   ;returns to X windows environment
 set_plot,'WIN'   ;returns to X windows environment
 print,string(10b),'*** Postscript file created as C:\idl.ps ***'
 i=ilast
endif

if i eq -7 then begin   ;use flat fields
 ff=-ff
 i=ilast
endif

if i eq -8 then begin   ;create jpeg image
 pout=bytscl(p,min=min,max=max,top=255)
 outfile='C:/Image'+sstr(ilast)+'.jpg'
 check_file=file_search(outfile,count=nfiles)
 ans=''
 if nfiles ne 0 then begin
  read,outfile+' Exists ***, Replace?? ', ans
  if ans ne 'y' then goto,skip_write
 endif
```

```
  write_jpeg,outfile,pout,quality=100
  print,'*** Made jpeg image under C: ***'
  skip_write:
  i=ilast
 endif

; if i eq -8 then begin   ;create png image
;   pout=p*21845.0/mean(p)  ;scale average up to 1/3 of 16 bits
;   sigma=stdev(pout,avep)
;   index=where(pout gt avep+4.0*sigma)
;   ;;print,min(pout),avep,max(pout),sigma
;   if index(0) eq -1 then goto,skip2
;   pout(index)=max(pout(where(pout le avep+4.0*sigma))) ;chops off histogram above 4 sigma
;   skip2:
;   minpt=min*21845.0/mean(p)>0
;   maxpt=max*21845.0/mean(p)
;   p_png=uint(65535d0*(double(pout)-double(minpt))/(double(maxpt)-double(minpt)))
;   outfile='C:/Image'+sstr(ilast)+'.png'
;   write_png,outfile,p_png
;   print,'*** Made png image under C: ***'
;  i=ilast
; endif

if i eq -10 then stop   ;for debugging


 if i le 0 or i gt nfiles then begin    ;for i input not in range
  print,'Invalid value'
  goto,menu
 endif


;_____________________________________________________________________________
;Read & present file...

 d_read,log+db+'\Image\'+files(i-1),h,p

 ;determine image type and apply flat field if required...
 type=d_value(h,96)
 typ='Unknown'
```

```
model=d_value(h,219)   ;the model is needed to get the right flat field...
case model of
 'DISR1': model='DISR#1'
 'DISR2': model='DISR#2'
 'DISR3': model='DISR#3'
 else: model='Unknown'
 endcase

 if model eq 'Unknown' then begin
  print,'There is no flat field for model: ',d_value(h,219)
  ff=-1
 endif

if type eq 21 then begin
 typ='MRI '
  if ff eq 1 then begin
   ;d_read,'\local\Imagers\Flat_Field\'+model+'\dli2_flat_field.disrsoft',hff,pff
   d_read,'C:\idl_programs\Flat_Field\'+model+'\dli2_flat_field.disrsoft',hff,pff
   if model eq 'DISR#1' then pff=pff/256.  ;DISR1 Flat field is wrong
   p=p*pff       ;removes flat field
  endif
endif
if type eq 22 then begin
 typ='SLI '
  if ff eq 1 then begin
   ;d_read,'\local\Imagers\Flat_Field\'+model+'\sli_flat_field.disrsoft',hff,pff
   d_read,'C:\idl_programs\Flat_Field\'+model+'\sli_flat_field.disrsoft',hff,pff
  if model eq 'DISR#1' then pff=pff/256.   ;DISR1 Flat field is wrong
  p=p*pff        ;removes flat field
 endif
endif
if type eq 23 then begin
 typ='HRI '
 if ff eq 1 then begin
  ;d_read,'\local\Imagers\Flat_Field\'+model+'\dli1_flat_field.disrsoft',hff,pff
  d_read,'C:\idl_programs\Flat_Field\'+model+'\dli1_flat_field.disrsoft',hff,pff
  if model eq 'DISR#1' then pff=pff/256.   ;DISR1 Flat field is wrong
  p=p(*,2:255)*pff           ;removes flat field
 endif
endif
```

```
 ilast=i
 p2=rotate(p,2)         ;rotate image 180 degrees so that up is up (p2).
 s=size(p2)
 ;p3=rebin(p2,2*s(1),2*s(2))   ;,/sample)       ;doubles image size
 x=2*s(1) & y=2*s(2)
 p3=congrid(p2,x,y,/center,cubic=-0.5,/interp)       ;doubles image size (p3)
 erase
 if auto eq 2 then begin & min=min(p) & max=min([2*mean(p),4095]) & endif
 if auto eq 1 then begin & min=min(p) & max=max(p) & endif

 ;tv,bytscl(p2,min=min,max=max,top=255)   ;to make actual size jpegs
 tv,bytscl(p3,min=min,max=max,top=255)


 print,string(10b),cdir,'\Log\',db,'\',files(ilast-1)
 print,typ,sstr(i),'  Min= ',sstr(min(p)),', Ave= ',sstr(mean(p)),', Max= ',sstr(max(p))
 print,'Exposure (ms) = ',sstr(d_value(h,36)),'   lamps = ',sstr(d_value(h,102))
 if ff eq -1 then print,'Model= ',model,'  Flat Field is included' $
  else print,'Model= ',model,'  Flat Field is removed'
 print,'Streach Settings = ',min,max

goto, again

;____________________________________________________________________________
;Done...

exit:
wdelete,0
print,'Done'
cd,cdir
end
```

Example program #2: Summary Listing of IR spectrometer data...

```
pro ir_list_files,first

; This program prints out a summary list of the ir spectrometer measurements
; in the current directory.  It is to be run from the test log (i.e. the directory containing "DB").
; 'first' is the number of the first IR to read (default = 0)
; The output is place in the log as IR_list.out

if n_params() eq 0 then first = 0  ;i.e. begin with first IR in Log

cd,'', current=cdir & cd,cdir      ;make cdir the test log directory
cd,cdir+'/DB2/Ir'

files=findfile('*_Ir',count=nfiles)

; Output data to root directory as IR_list.out
close,1
openw,1,'C:\IR_list.out'
printf,1,cdir
printf,1,string(10b),'There are ',sstr(nfiles),' IR files in this Log.'
printf,1,' * = Possibly Saturated Data, min(measurement) < 15000 dn'

printf,1,''
printf,1,'   N   IR    Time  altitude  Cycle  Rows  Type           Ave.     Max.   Lamp     Exp.   Collect
Rots Regns Min(p)   Temp.    Resol.   1 Rot.
printf,1,'        #     min     Km       #                          dn       dn             ms       s
dn        K        Km       Km
printf,1,''

nn=0  ; files counter

for i=first,nfiles-1 do begin
 d_irread,files(i),h,p,g
 siz=size(p)
 seq=d_value(h,87)            ;measurement sequence number
 time=d_value(h,88)           ;mission time in seconds
 temp=(d_value(h,105)+d_value(h,106))/2.        ;average IR temperature during measurement
 collect=d_value(h,108)       ;collection time in seconds
```

```
 rows=d_value(h,7)              ;number of data rows (spectra)
 cycle=d_value(h,89)            ;cycle number
 alt=d_value(h,98)/1000.        ;altitude in Km
 lamps=''
 lamps=d_value(h,102)           ;lamp state (i.e. 0000)
 g1=g.(1)                       ;Sample-time from structure g
 rots=d_value(h,109)            ;number of rotations
 regions=d_value(h,110)         ;number of IR regions
 exp=g1(5)*8.064
 pi=3.141592654
 tt=tan(24.5*pi/180)-tan(15.5*pi/180)     ;ratio of altitude to anulis thickness 9 deg at 20 deg
 cir=pi*tan(20*pi/180)
 sat=''
 type=mtype(files(i))
 typ='Unknown'
 if type eq 8 then typ='DLIS'
 if type eq 9 then typ='ULIS'
 if type eq 10 then typ='Comb'
 if type eq 11 then typ='Long   '

;for descent ir's...
if siz(2) eq 24 then begin
 r=p(*,8:15)-p(*,0:7)   ;DLIS Closed-Open
 aved=intarr(8)
 for j=0,7 do aved(j)=mean(r(9:140,j))
 if min(p(9:140,*)) le 15000 then sat=' * ' else  sat='   '
 for j=0,7 do
printf,1,nn,seq,time/60.,alt,cycle,rows,typ,j,aved(j),max(r(9:140,j)),lamps,exp,collect,rots,regions,min(p(
9:140,0:7)),sat,temp,alt*tt,alt*cir, $
  format='(i4,i5,f9.2,f9.2,i5,i7,3x,a," D-",i2,T54,f10.2,i8,2x,a4,f9.1,
f10.2,2x,i4,2x,i3,i7,a3,f7.1,f11.2,f11.2)'

 r=p(*,20:23)-p(*,16:19)        ;ULIS Closed-Open
 aveu=intarr(4)
 for j=0,3 do aveu(j)=mean(r(9:140,j))
 if min(p(9:140,*)) le 15000 then sat=' * ' else  sat='   '
 for j=0,3 do
printf,1,nn,seq,time/60.,alt,cycle,rows,typ,j,aved(j),max(r(9:140,j)),lamps,exp,collect,rots,regions,min(p(
9:140,0:7)),sat,temp,alt*tt,alt*cir, $
  format='(i4,i5,f9.2,f9.2,i5,i7,3x,a," U-",i2,T54,f10.2,i8,2x,a4,f9.1,
f10.2,2x,i4,2x,i3,i7,a3,f7.1,f11.2,f11.2)'
```

```
  goto,next_file
endif

;for simple ulis or dlis files
 r=p(*,1)-p(*,0)             ;Shutter Closed-Open.
 if min(p(9:140,0)) le 15000 then sat=' * ' else  sat='   '
 if type eq 10 then typ2=typ+' dn' else typ2=typ+'   '

printf,1,nn,seq,time/60.,alt,cycle,rows,typ,mean(r(9:140,*)),max(r(9:140,*)),lamps,exp,collect,rots,regions
,min(p(9:140,0)),sat,temp,alt*tt,alt*cir, $
  format='(i4,i5,f9.2,f9.2,i5,i7,3x,a,T54,f10.2,i8,2x,a4,f9.1, f10.2,2x,i4,2x,i3,i7,a3,f7.1,f11.2,f11.2)'

;for ircomb files...
if siz(2) eq 4 then begin
 if min(p(9:140,2)) le 15000  then sat=' * ' else  sat='   '
 r=p(*,3)-p(*,2)              ;Upward looking data for Combined reads
 if type eq 10 then typ2=typ+' up'

printf,1,nn,seq,time/60.,alt,cycle,rows,typ,mean(r(9:140,*)),max(r(9:140,*)),lamps,exp,collect,rots,regions
,min(p(9:140,2)),sat,temp,alt*tt,alt*cir, $
  format='(i4,i5,f9.2,f9.2,i5,i7,3x,a,T54,f10.2,i8,2x,a4,f9.1, f10.2,2x,i4,2x,i3,i7,a3,f7.1,f11.2,f11.2)'
endif


next_file:
nn=nn+1
endfor

exit:
cd,cdir
print,cdir
print,'Done!'
end
```

Details of IDL_PROGRAMS directory...

The following is a listing of the subdirectories contained in the IDL_PROGRAMS directory in the DISR archive. The first number after the directory name is the number of subdirectories contained in that directory, and the second number is the quantity of files at that level.

IDL_PROGRAMS (8 - 0)

-> BASHAR_RIZK (15 - 0)
--|| Bashar (0 - 2110)
--|| Prep Changes 1-14-05 (0 - 3)
--|| br (0 - 3)
--|| cpe1 (0 - 21)
--|| gtsl1 (0 - 22)
--|| hp (0 - 14)
--|| m4 (0 - 44)
--|| merge (0 - 1)
--|| oc (0 - 68)
--|| other (0 - 13)
--|| prep (0 - 41)
--|| specmap (0 - 52)
--|| st2 (0 - 80)
--|| supersim (0 - 12)
--|| table (0 - 11)

-> CHUCK_SEE (22 - 0)
--|| 2ndary (0 - 1)
--|| Backups (0 - 3)
--|| Erich's Kabs 2005 (1 - 0)
---||| Software (1 - 0)
----|||| Spectra (0 - 41)
--|| PDS (8 - 29)
---||| BROWSE (2 - 1200)
----|||| IMG (0 - 0)
----|||| PNG (0 - 1200)
---||| DATA (13 - 0)
----|||| DARK (2 - 0)
----|||| DESCENT (2 - 0)
----|||| HKEEPING (1 - 0)
----|||| IMAGE (13 - 0)
----|||| IR (3 - 0)
----|||| LAMP (10 - 0)
----|||| SOLAR (2 - 0)
----|||| STRIP (2 - 0)
----|||| SUN (1 - 0)
----|||| TIME (4 - 0)
----|||| VIOLET (7 - 0)
----|||| VISIBLE (11 - 0)

----|||| VISIBLE_EXT (11 - 0)
---||| DDP (7 - 1)
----|||| DLIS (0 - 6)
----|||| DLV (0 - 5)
----|||| DLVS (0 - 2)
----|||| Old (0 - 8)
----|||| ULIS (0 - 6)
----|||| ULV (0 - 6)
----|||| ULVS (0 - 2)
---||| DERIVED_DATA_PRODUCTS (6 - 0)
----|||| DLIS (0 - 6)
----|||| DLV (0 - 4)
----|||| DLVS (0 - 2)
----|||| ULIS (0 - 6)
----|||| ULV (0 - 4)
----|||| ULVS (0 - 2)
---||| Junk (0 - 2)
---||| Old_programs (1 - 6)
----|||| PDS_23Apr06 (0 - 20)
---||| Out (0 - 43)
---||| idl_programs (0 - 17)
--|| SA_simulator (4 - 66)
---||| Results (0 - 4)
---||| Source and Unused Files (2 - 29)
----|||| from lyn (0 - 17)
----|||| to lyn (0 - 2)
---||| picture of board equations approach (0 - 12)
---||| test cases (1 - 9)
----|||| disr3_cal.24Jul96.absresp_cold_24Jul96.4 (0 - 5)
--|| Surf_Images (0 - 6)
--|| UNIX (10 - 0)
---||| 2ndary (0 - 1)
---||| PB_160 (1 - 1)
----|||| Programs (6 - 94)
---||| cmdndata (0 - 17)
---||| data (0 - 59)
---||| des_cal_cycles (0 - 13)
---||| hc_comp (0 - 20)
---||| hkeeping (0 - 34)
---||| ptest (0 - 79)
---||| trash (0 - 2)
---||| util (0 - 68)
--|| VIMS (1 - 1)
---||| idl (5 - 3)
----|||| HST (6 - 34)
----|||| VIMS data (0 - 17)
----|||| VIMS_ir_Aug2007 (1 - 0)
----|||| VIMS_vis_Aug2007 (1 - 0)
----|||| namlists (1 - 8)
--|| cmdndata (0 - 17)
--|| data (0 - 138)

--|| des_cal_cycles (0 - 13)
--|| hc_comp (0 - 20)
--|| hkeeping (0 - 35)
--|| ir (0 - 0)
--|| movie_complete (1 - 222)
---||| jpegs (0 - 103)
--|| other (0 - 75)
--|| ptest (0 - 78)
--|| radxfer (0 - 2)
--|| temp (0 - 0)
--|| thn (0 - 7)
--|| trash (0 - 2)
--|| util (0 - 81)

-> LISA_MCFARLANE (4 - 0)
--|| Archive_junk (0 - 12)
--|| Lisa Programs (1 - 1)
---||| PDS_prog_20Mar04 (1 - 14)
----|||| other (0 - 1)
--|| Lyn programs (0 - 6)
--|| PDS_prog_Feb04 (0 - 19)

-> LYN_DOOSE (3 - 1)
--|| IR Simulator (0 - 94)
--|| Lyns (0 - 1)
--|| keycode (0 - 3)

-> MICHAEL_BUSHROE (2 - 1)
--|| Mike (0 - 3)
--|| old (2 - 0)
---||| batch (6 - 0)
----|||| gse (0 - 8)
----|||| new (3 - 0)
----|||| old (0 - 169)
----|||| panoramas (0 - 14)
----|||| sharpening (0 - 4)
----|||| specmap (0 - 4)
---||| telemetry (0 - 8)

-> MICHAEL_KUEPPERS (1 - 0)
--|| SOFTWARE (0 - 6)

-> OTHER (3 - 0)
--|| CAL_SOFT (3 - 3)
---||| disrsoft (14 - 4)
----|||| batch (0 - 11)
----|||| common (0 - 18)
----|||| doc (0 - 14)
----|||| enhance (0 - 2)
----|||| local (0 - 62)
----|||| output (0 - 16)

----|||| radio (0 - 4)
----|||| readme (0 - 8)
----|||| simulate (0 - 6)
----|||| startup (0 - 7)
----|||| system (1 - 46)
----|||| tools (0 - 76)
----|||| verify (0 - 12)
----|||| xy (0 - 1)
---||| journal (0 - 275)
---||| win32 (14 - 0)
----|||| batch (0 - 12)
----|||| common (0 - 15)
----|||| data (0 - 0)
----|||| doc (0 - 11)
----|||| enhance (0 - 3)
----|||| output (0 - 16)
----|||| radio (0 - 5)
----|||| readme (0 - 9)
----|||| simulate (0 - 7)
----|||| startup (0 - 6)
----|||| system (0 - 46)
----|||| tools (0 - 78)
----|||| verify (0 - 13)
----|||| xy (0 - 2)
--|| FLAT_FIELD (3 - 1)
---||| DISR#1 (0 - 3)
---||| DISR#2 (0 - 3)
---||| DISR#3 (0 - 16)
--|| NEW_GSE (0 - 28)

-> STEFFI_ENGEL (54 - 13)
--|| AZ comparison (0 - 9)
--|| AZ ir (0 - 19)
--|| Copy of DLVS-ULVS before combining corr+fluxes (4 - 1)
---||| DN-Fluxes (4 - 9)
----|||| input (1 - 6)
----|||| new dark (0 - 13)
----|||| old (0 - 2)
----|||| output (2 - 4)
---||| VIS_corr_factors (3 - 30)
----|||| input (1 - 39)
----|||| not needed (0 - 14)
----|||| output (0 - 10)
---||| input (0 - 6)
---||| old (3 - 0)
----|||| DLVS (3 - 10)
----|||| ULVS (2 - 11)
----|||| VIS_corr_factors_old (2 - 43)
--|| DATA (4 - 0)
---||| Descent_test_11Sep96.2 (3 - 26)
----|||| DB (27 - 0)

----|||| Verification (2 - 0)
----|||| post (2 - 0)
---||| Descent_test_11Sep96.2_sim (2 - 1)
----|||| DB (5 - 0)
----|||| post (1 - 0)
---||| Descent_test_11Sep96.3 (3 - 18)
----|||| DB (26 - 0)
----|||| Verification (2 - 0)
----|||| post (2 - 0)
---||| Descent_test_11Sep96.3_sim (3 - 4)
----|||| DB (6 - 0)
----|||| Dark (0 - 81)
----|||| post (1 - 0)
--|| DLIS-ULIS (10 - 17)
---||| DN-fluxes (2 - 11)
----|||| corr_factors (2 - 0)
----|||| input (0 - 34)
---||| DN-fluxes_jan18 (6 - 17)
----|||| corr_factors (8 - 6)
----|||| good copy (0 - 0)
----|||| input (0 - 34)
----|||| last changes (0 - 17)
----|||| old (0 - 4)
----|||| output (11 - 11)
---||| DN-fluxes_old (2 - 4)
----|||| input (0 - 12)
----|||| output (0 - 6)
---||| IR_corr_factors (3 - 13)
----|||| input (0 - 31)
----|||| output (1 - 0)
----|||| probably don't need (0 - 2)
---||| analysis (3 - 11)
----|||| jan 11 (0 - 6)
----|||| jan12 (0 - 2)
----|||| old (0 - 5)
---||| input (1 - 1)
----|||| old (0 - 2)
---||| output (2 - 10)
----|||| old (0 - 18)
----|||| old1 (0 - 9)
---||| plots (2 - 8)
----|||| old (0 - 4)
----|||| old1 (0 - 4)
---||| titan14 (1 - 16)
----|||| jan14 data (0 - 10)
---||| titan14_jan19 (0 - 21)
--|| DLIS_descent (2 - 4)
---||| input (0 - 19)
---||| output (0 - 21)
--|| DLIS_descent_Apr25 (3 - 10)
---||| April 2005 (0 - 19)

---||| input (0 - 9)
---||| output (0 - 11)
--|| DLIS_descent_Dec_2005 (3 - 6)
---||| input (1 - 9)
----|||| old (0 - 10)
---||| old (0 - 1)
---||| output (2 - 20)
----|||| old (0 - 9)
----|||| old up to may 2006 (0 - 42)
--|| DLVS-ULVS (10 - 4)
---||| DN-Fluxes (5 - 28)
----|||| corr_factors (2 - 8)
----|||| input (1 - 32)
----|||| misc (0 - 8)
----|||| old subroutines (0 - 8)
----|||| output (1 - 7)
---||| DN-Fluxes_jan19 (5 - 20)
----|||| corr_factors (4 - 5)
----|||| input (1 - 32)
----|||| misc (0 - 8)
----|||| old subroutines (0 - 8)
----|||| output (12 - 14)
---||| DN-Fluxes_single_spectra (0 - 13)
---||| VIS_corr_factors (3 - 22)
----|||| input (1 - 21)
----|||| not needed (0 - 17)
----|||| output (1 - 1)
---||| analysis (3 - 18)
----|||| input (1 - 6)
----|||| old (0 - 7)
----|||| output (0 - 2)
---||| input (1 - 13)
----|||| old (0 - 2)
---||| misc (1 - 17)
----|||| plots for talks (0 - 13)
---||| old (3 - 0)
----|||| DLVS (3 - 10)
----|||| ULVS (2 - 11)
----|||| VIS_corr_factors_old (2 - 43)
---||| output (7 - 20)
----|||| old (0 - 23)
----|||| old2 (0 - 6)
----|||| old3 (0 - 14)
----|||| old4 (0 - 12)
----|||| old5 (0 - 9)
----|||| old6 (0 - 24)
----|||| old7 (0 - 12)
---||| plot visible (2 - 6)
----|||| namelist files (0 - 4)
----|||| old (0 - 9)
--|| DLVS_4avg_sc (1 - 9)

---||| output (2 - 27)
----|||| final output (1 - 41)
----|||| some output (0 - 26)
--|| DLVS_check_wavelength_cal (0 - 6)
--|| DLVS_descent (4 - 14)
---||| crosstalk (1 - 23)
----|||| probably no use (0 - 1)
---||| input (0 - 10)
---||| output (3 - 14)
----|||| Feb7 (0 - 6)
----|||| March8 (0 - 5)
----|||| extra (1 - 7)
---||| responsitivities (0 - 9)
--|| DLVS_descent_3_14_2005 (2 - 14)
---||| input (0 - 8)
---||| output (2 - 11)
----|||| Feb7 (0 - 6)
----|||| extra (1 - 6)
--|| DN_sec (0 - 4)
--|| Erich_absorption_coef (9 - 34)
---||| Marty talks (0 - 4)
---||| Radiative transfer for PC (0 - 1)
---||| analysis (2 - 23)
----|||| sept20,06 (0 - 17)
----|||| sept6,06 (0 - 29)
---||| dont need (0 - 1)
---||| esfit1 (0 - 5)
---||| excel (0 - 26)
---||| input (3 - 18)
----|||| old (0 - 3)
----|||| sept20,06 (0 - 11)
----|||| sept6,06 (0 - 28)
---||| ir_model (1 - 56)
----|||| Fortran (0 - 11)
---||| rotation_function (2 - 11)
----|||| old programs (0 - 1)
----|||| output (0 - 116)
--|| Far_IR (3 - 14)
---||| Rad_Transfer (5 - 4)
----|||| analysis (0 - 8)
----|||| input (0 - 5)
----|||| model (1 - 764)
----|||| output (0 - 11)
----|||| read_intensi (1 - 6)
---||| Rad_Transfer_VIMS (6 - 9)
----|||| VIMS spectra (0 - 6)
----|||| analysis (0 - 3)
----|||| input (0 - 12)
----|||| model (0 - 342)
----|||| output (0 - 11)
----|||| read_intensi (0 - 6)

---||| methane_coeff (0 - 12)
--|| Heating Rates (3 - 18)
---||| Rad_Transfer (4 - 3)
----|||| input (0 - 5)
----|||| model (1 - 196)
----|||| output_paper1 (1 - 68)
----|||| read_intensi (0 - 4)
---||| graphs for heating paper (0 - 34)
---||| heating (1 - 23)
----|||| older programs (0 - 1)
--|| Heating Rates2 (3 - 0)
---||| Rad Transfer (4 - 6)
----|||| input (1 - 6)
----|||| model (0 - 114)
----|||| output (6 - 0)
----|||| probably don't need (0 - 1)
---||| Rad Transfer vis (3 - 6)
----|||| input (0 - 4)
----|||| model (2 - 224)
----|||| output (4 - 0)
---||| heating (2 - 3)
----|||| output (0 - 9)
----|||| plots (0 - 8)
--|| IR (10 - 1)
---||| IR alt cont models (3 - 6)
----|||| analysis (6 - 14)
----|||| ir_model (1 - 5)
----|||| rotation_function (2 - 9)
---||| IR_corr_factors (4 - 12)
----|||| 2nd try (0 - 16)
----|||| input (2 - 10)
----|||| output (6 - 0)
----|||| probably don't need (0 - 3)
---||| IR_data (2 - 0)
----|||| Descent_test_11Sep96.2 (3 - 26)
----|||| Descent_test_11Sep96.2_sim (2 - 1)
---||| Radiative Transfer (0 - 7)
---||| analysis (5 - 10)
----|||| Lyn_fluxes (0 - 5)
----|||| input (0 - 5)
----|||| old (0 - 12)
----|||| old2 (0 - 2)
----|||| tauwagg (3 - 7)
---||| input (0 - 0)
---||| old (1 - 3)
----|||| IR_Simulator_old (3 - 14)
---||| old stuff (2 - 22)
----|||| old (2 - 0)
----|||| old programs (0 - 4)
---||| output (0 - 0)
---||| read_ulis_dlis (5 - 7)

----|||| data1 (0 - 10)
----|||| data2 (0 - 6)
----|||| input (0 - 19)
----|||| old2 (0 - 5)
----|||| old3 (0 - 6)
--|| IR Simulator_Lyn (1 - 208)
---||| output (0 - 0)
--|| IR alt spec models (3 - 1)
---||| analysis (0 - 16)
---||| ir_model (0 - 6)
---||| rotation_function (2 - 9)
----|||| input (0 - 6)
----|||| output (0 - 12)
--|| IR alt spec models 2006 (11 - 13)
---||| Bruno methane coeff (0 - 6)
---||| analysis (4 - 42)
----|||| october06 (0 - 19)
----|||| summer06 (0 - 24)
----|||| summer06_b (0 - 8)
----|||| wrong (0 - 2)
---||| input_fall06 (1 - 26)
----|||| oct20006 (0 - 9)
---||| ir_model (3 - 187)
----|||| october06 (0 - 143)
----|||| summer06 (0 - 13)
----|||| summer06_b (0 - 3)
---||| kabsfront (2 - 38)
----|||| Lyn's versions (0 - 3)
----|||| zip (0 - 1)
---||| model input (0 - 25)
---||| phase_function (0 - 2)
---||| rotation_function (1 - 10)
----|||| output (1 - 38)
---||| summer06 (0 - 32)
---||| summer07 (0 - 10)
---||| test_input (1 - 37)
----|||| lyn files (0 - 2)
--|| IR analysis Aug05 (6 - 15)
---||| calculate_weights (2 - 2)
----|||| input (0 - 2)
----|||| output (0 - 3)
---||| calculate_weights_dlis (1 - 4)
----|||| output (0 - 3)
---||| comparison (1 - 4)
----|||| input (0 - 3)
---||| ir_models (2 - 9)
----|||| extra (0 - 9)
----|||| old (0 - 4)
---||| rotation_function (3 - 10)
----|||| input (0 - 22)
----|||| old (0 - 2)

----|||| output (1 - 0)
---||| rotation_function_dlis (2 - 10)
----|||| input (2 - 19)
----|||| output (2 - 38)
--|| IR analysis Nov05 (5 - 14)
---||| comparison (1 - 3)
----|||| input (0 - 6)
---||| ir_models2 (1 - 15)
----|||| old (0 - 6)
---||| ir_models_old (0 - 8)
---||| rotation_function (4 - 9)
----|||| backup Jan 2006 (0 - 9)
----|||| input (1 - 2)
----|||| input_files (0 - 2)
----|||| output (3 - 6)
---||| smooth_ir (3 - 15)
----|||| input (0 - 2)
----|||| input_files (0 - 8)
----|||| output (0 - 46)
--|| IR analysis Oct05 (5 - 5)
---||| calculate_weights_dlis (1 - 3)
----|||| output (0 - 7)
---||| calculate_weights_ulis (1 - 2)
----|||| output (0 - 10)
---||| comparison (1 - 2)
----|||| input (0 - 4)
---||| ir_models (0 - 10)
---||| rotation_function (4 - 9)
----|||| input (5 - 25)
----|||| input_files (0 - 2)
----|||| old programs (0 - 1)
----|||| output (5 - 50)
--|| IR combined (5 - 23)
---||| Radiative transfer for PC (0 - 1)
---||| analysis (0 - 16)
---||| ir_model (1 - 49)
----|||| Fortran (0 - 12)
---||| rotation_function (2 - 11)
----|||| old programs (0 - 1)
----|||| output (1 - 40)
---||| summer06 (0 - 4)
--|| IR combined with amoeba (5 - 13)
---||| Radiative transfer for PC (0 - 1)
---||| analysis (0 - 16)
---||| ir_model (1 - 28)
----|||| Fortran (0 - 11)
---||| output (1 - 10)
----|||| old 1 (0 - 15)
---||| rotation_function (2 - 10)
----|||| old (0 - 1)
----|||| output (0 - 26)

--|| IR ground cont models (3 - 1)
---||| analysis (2 - 6)
----|||| feb06 (0 - 16)
----|||| jan30 (0 - 11)
---||| ir_model (0 - 6)
---||| rotation_function (2 - 9)
----|||| input (2 - 3)
----|||| output (2 - 7)
--|| IR models (0 - 7)
--|| IR rot (3 - 18)
---||| input (0 - 18)
---||| output (0 - 8)
---||| probably don't need (0 - 27)
--|| Stefan (3 - 21)
---||| IR (0 - 40)
---||| don't need this (0 - 3)
---||| input (0 - 6)
--|| ULIS_descent (2 - 4)
---||| input (0 - 23)
---||| output (0 - 12)
--|| ULIS_descent_Apr25 (2 - 20)
---||| input (0 - 21)
---||| output (0 - 15)
--|| ULIS_descent_Dec_2005 (2 - 4)
---||| input (1 - 15)
----|||| old (0 - 10)
---||| output (1 - 24)
----|||| old (0 - 2)
--|| ULVS (4 - 10)
---||| 4avg (2 - 8)
----|||| extra copy (0 - 1)
----|||| output (3 - 6)
---||| output (1 - 3)
----|||| final output (0 - 21)
---||| ulvs_single_col (0 - 6)
---||| xls files (0 - 1)
--|| ULVS_descent (4 - 36)
---||| input (1 - 22)
----|||| corr_files (0 - 5)
---||| make crosstalk factors (0 - 14)
---||| output (5 - 23)
----|||| Feb18 (0 - 13)
----|||| Feb22 (0 - 8)
----|||| extra (1 - 18)
----|||| old aug05 (0 - 27)
----|||| old feb7 (0 - 4)
---||| responsitivities (0 - 9)
--|| ULVS_descent_feb16 (3 - 21)
---||| input (0 - 10)
---||| make crosstalk factors (0 - 14)
---||| output (2 - 11)

----|||| extra (0 - 8)
----|||| old feb7 (0 - 4)
--|| ULVS_descent_feb9 (2 - 14)
---||| input (0 - 1)
---||| output (1 - 8)
----|||| extra (0 - 7)
--|| VIMS data (0 - 21)
--|| VIMS models (0 - 21)
--|| VIMS_ir_Aug2007 (1 - 8)
---||| idl (3 - 7)
----|||| input (0 - 15)
----|||| model (0 - 339)
----|||| output (1 - 23)
--|| VIMS_vis (4 - 11)
---||| input (2 - 22)
----|||| Normalized phase functions 5-7-2007 (0 - 15)
----|||| old (0 - 3)
---||| model (0 - 223)
---||| output (0 - 9)
---||| paulo (2 - 4)
----|||| input (0 - 6)
----|||| model (0 - 6)
--|| VIMS_vis_Aug2007 (3 - 13)
---||| VIMS_TB_Huygens (0 - 21)
---||| idl (3 - 16)
----|||| input (1 - 12)
----|||| model (1 - 244)
----|||| output (1 - 26)
---||| more of the same (1 - 1)
----|||| VIMS_TB_1 (0 - 21)
--|| crosstalk (8 - 0)
---||| dlvs (5 - 0)
----|||| input (0 - 21)
----|||| maybe (0 - 3)
----|||| output (0 - 2)
----|||| reduction (1 - 15)
----|||| zenith (3 - 13)
---||| excel (0 - 16)
---||| input (1 - 22)
----|||| old input (0 - 2)
---||| input_programs (2 - 18)
----|||| Lyn (0 - 10)
----|||| old (0 - 4)
---||| old programs (0 - 2)
---||| old programs 2 ulvs (0 - 14)
---||| ps (0 - 2)
---||| ulvs (4 - 1)
----|||| descent (0 - 6)
----|||| find_factor (0 - 20)
----|||| reduction (3 - 13)
----|||| useless (0 - 1)

--|| descent_input (0 - 22)
--|| dlis_ulis_disr2 (1 - 10)
---||| output (0 - 4)
--|| input (0 - 15)
--|| matrix (3 - 15)
---||| excel (0 - 11)
---||| input (0 - 2)
---||| original (0 - 6)
--|| more_programs (1 - 7)
---||| convert_xls (0 - 8)
--|| namlists (0 - 7)
--|| old_programs (7 - 7)
---||| Oct10,02 (0 - 6)
---||| Sept11,02 (0 - 2)
---||| Sept12,02 (0 - 5)
---||| Sept18,02 (0 - 15)
---||| Sept25,02 (0 - 1)
---||| Sept4,02 (0 - 5)
---||| Sept9,02 (0 - 2)
--|| output (2 - 5)
---||| save output (0 - 27)
---||| sorting (0 - 24)
--|| sep25 old (2 - 0)
---||| 2ndtry (0 - 2)
---||| they work!! (0 - 2)
--|| test_tip (0 - 27)