

GeoPIXE Change Log

GeoPIXE-open-source

Note: For complex changes, tag the source file lines with: ;@month-year (e.g. ;@3-16).

6 Feb, 2023

1. GitHub
 - a. Initial load of GeoPIXE to local Git and to GeoPIXE-org on GitHub at URL:
 - b. <https://github.com/CSIRO-GeoscienceAnalytics/GeoPIXE-org> as branch “main”.
2. GeoPIXE
 - a. Image_table
 - i. Look for GUI plugins ('image_table_*_gui_plugin.sav') and add these to the 'tab_panel'.
 - b. Image_table_eventcb
 - i. OnButton_Image_Table_Export
 1. Uses event.top to get the pstate.
 - ii. Any that need hotspot_element, use minimal_centroid_element
 - iii. OnButton_Image_Table_Modify
 1. use minimal_centroid_element, test valid.
 - c. Rename “image.pro” to “gimage.pro” and main routine, xmanager and events.
 - i. Duplicate in IDL88/lib/graphics.
 - ii. Occurrences: gimage, daq_launch, spectrum_display, maia_launch
 - d. Version 8.8a.
3. image_table_xfm_gui_plugin
 - a. Add a plugin to add to Image_Table tabs for the XFM beamline functions.
 - i. Uses a local pstate, which points to the Image-Table pstate_parent.
 - ii. Event routines (copied from image_table) set 'pstate' to the parent pstate and pstate_local to the local one, which makes the code work with little change.
4. Builder
 - a. Add build for GUI plugins.
5. Falconx_device
 - a. Get_header_info method
 - i. Do not strip trailing “_0” from ‘output’, as it’s not there.
 - ii. Accept a single line ylut table (e.g. for a single .silist file).

1 Feb, 2023

1. Eclipse “clean up”
 - a. Move .log and log files to “old/old-files-Open-Source” directory, from:
 - i. .metadata
 - ii. .metadata/.plugins/org.eclipse.ui.workbench
 - b. Move the following dirs with history to “old” as well:
 - i. .metadata/.plugins/org.eclipse.core.resources/.history
 - ii. .metadata/.plugins/org.eclipse.ltk.core.refactoring/.refactorings
2. GeoPIXE
 - a. File_requester
 - i. If geopixe_root does not exist make it blank string.
 - b. Read_detector
 - i. If ‘layout’ file does not exist, try same path as detector file ‘f’.
 - c. Fit_setup/load_pcm_parameters
 - i. Rearrange to better use ‘detector_update’ results.
 - ii. Only popup warning if detector not found.
 - d. Depth_ratio_yields
 - i. Use layout as is, now that read_detector fixes path.

3. Depth wizard
 - a. wizard_depth_calc_depth_curve
 - i. Use layout as is, now that read_detector fixes path.

21 Dec, 2022

1. builder (NEW)
 - a. Build a script “build.spro” to be used (use: “@build.spro”) to selectively build the SAV files, by categories ('all', 'plugin', 'back', 'spectrum', 'image', 'device', 'wizard', 'source', 'maia', 'daq', 'main', 'browse']).
 - i. This avoids issue with poor control of !PATH and using ‘.compile’ by file-name, rather than ‘resolve_routine’.
 - b. Does .full_reset_session for each project build to clear compiled routines.
 - i. But this makes it very slow if run from IDLDE.
 - c. !PATH built afresh for each project build to include whole “main” tree plus the project.
 - d. Build_project:
 - i. Does the fiddly work for builder.
 - e. Build_database
 - i. Does the database build part.
 - f. The following builds do a resolve_all:
 - i. Main, IDL Query, GeoPIXE update and Builder.
 - ii. IDL Query now does not need the make routine anymore.
 - iii. GeoPIXE update of a non-GUI version is not done yet.
2. projects:
 - a. Needed to enforce consistency in naming:
 - i. SAV file name constructed from project name, with additional options to tag a suffix to SAV filename.
 - ii. Build from Eclipse must match this SAV filename.
 - iii. Deleted some projects not needed for open source, such as junk tests, etc.
 - b. image_table_evt
3. Organization
 - a. The “builder” imposed some rules that needed to be enforced:
 - i. Plugins must start with “Back”, “Spectrum”, “Image” and end with “plugin”.
 - ii. Except “source” and “Wizards” plugins.
 - iii. Maia projects start with “Blog” or “Maia”.
 - iv. All Maia SAV end up in /maia, including maia_control.sav and maia_scan_list.sav.
 - v. All DAQ SAV end up in /daq, including daq_control.sav.
 - vi. All browsers start with “Browse”.
 - vii. Some get renamed: mm_scan_list → maia_scan_list, GeoPIXE worker → GeoPIXE parallel, “ftp update” → “GeoPIXE update”.
 - viii. ‘misc’ compiles/builds include ‘geopixe_index’, ‘geopixe_update’, “builder”, ...
 - ix. Fake Blogs divided into ET2 and DA2 versions.
 - b. Database build
 - i. ‘make_database.pro’ has changed, so it gets called from ‘builder’.
 - c. Build
 - i. Use ‘builder’ (pops up index selection) to select which classes to build, which creates “build.spro” (note: this will be in “geopixe” for runtime execution and “main” for IDLDE execution). A pop-up reminds the user of “build.spro”, its path and the need to run “@build.spro” from the IDL command line after a “cd” to path.
 - ii. “@build.spro” from IDL command line to build (after “cd” to the dir). This is much faster than in IDLDE due to long lag after a full reset.
4. GeoPIXE
 - a. Image_routines
 - i. Analyze_image
 1. Test on “uniform_element” should only be for non XANES image case.
 - b. Version 8.7o.

5. Maia
 - a. Version 8.7o.
6. DAQ
 - a. Version 8.7o.
7. Geopixe_update
 - a. Defaults to copy Maia files too.
 - b. Version 8.7o.

1 Dec, 2022

1. GeoPIXE
 - a. image_table_evt
 - i. Need to write combined spectra for this condition:
 1. repeat_loop **or** (*pstate).cluster
 - ii. Was not writing file for /cluster and no repeat.
 - b. save_plot_options
 - i. Save as version -5
 - ii. This should have been set before, with addition of Learn options.
 - c. image_routines
 - i. make_tvb
 1. accept 'colour' as vector for colour, dcolour (edge shading).
 - d. imagergb_routines
 - i. make_RGB_tvb (called 3x from 'make_RGB_true')
 1. Add 'centroid' and 'colour' keywords.
 2. default colour to white (255).
 3. Get qc and corr_on from pstate_parent
 4. Keep 'args' even though not implemented in keywords yet.
 5. Add centroids element detection.
 6. Add centroids circles into bc array
 7. Compress bc like b.
 8. Write bc above threshold into b
 9. Also test for "2" next to "0", which means a break in circle.
 - ii. make_RGB_true
 1. Add 'centroids' keyword.
 2. Pass this to make_RGB_tvb
 - e. plot_rgb_image_select
 - i. Added centroids controls and enable option button.
 - ii. Add options event for centroids (10).
 - iii. Add 'centroid-element-text' event
 - iv. Update 'plot.centroids.element' in Preview and OK
 - v. Add to defaults
 - vi. Update centroid on/element in plot_RGB_image_select_update
 - vii. save_plot_RGB_image_select saves as version -5
 1. Now uses 'save_plot_options'
 - ii. Preview mode – needs to temporarily disable Learn mode.
 - f. plot_rgb_images
 - i. Pass centroids to 'make_RGB_true'
 - ii. Needs to set image, image2, image3 in pstate to pass to make_RGB_tvb
 - g. plot_image_select
 - i. save_plot_image_select saves as version -5
 1. Now uses 'save_plot_options'
 - h. geopixe_do_commands
 - i. Add capacity for 'print_image_metadata' command
 - i. print_image_metadata (NEW)
 - i. Reads an image and outputs a metadata file
 - ii. uses "files=" and "output=" form for GCF approach

- iii. Has option for /stats
 - iv. Calls 'image_details' with /JSON
- j. image_history
 - i. Add a "C*" button to make a GCF file for metadata output.
 - ii. Add a "Output metadata" button.
- k. image_details
 - i. Use new routine 'add_history_item' to add items to the details output. Uses orderedhash for /JSON mode and string array normally.
 - ii. Found best to not use 'json_serialize' as it creates one long string.
 - iii. Used 'string(detail, /implied_print)' to get JSON like output, but in a string array, which makes it clearer, placed on separate lines.
- l. add_history_item' (NEW)
 - i. Manage adding new items rows to orderedhash or string array.
- m. find_device_objects
 - i. Remove reference to "now" in error catch, which is not used.
- n. export_images_rgb
 - i. Fixed bug that repeated element loop for Table mode export.
- o. OnNotify_image
 - i. 'batch-save' still had obsolete /tab – removed.
- 2. splash
 - a. Update years for ©
- 3. database
 - a. Build "geopixe2.sav", for splash update

10 Nov, 2022

- 1. !Path issue
 - a. Any attempt to set !PATH is overruled by open projects, even within an executing PRO.
 - b. This includes the subdirs "interface", "plugins", "wizards", which I tried to exclude using 'manage_path'.
 - c. It does not seem possible to manipulate !path adequately. Hence, we need to locate all compiled SAV files (and the "geopixe2.sav" database file) away from !path, and outside of the source dir tree (at least the "main" GeoPIXE tree).
 - d. Need to relocate "interface", "plugins", "wizards", "maia", "daq". Probably need to move "images", "Help".
 - e. This be done for all runtime data as well, including lib files, all detector, layout and filter files.
 - f. Change organization to use a "main" project as the main source tree and a new project "geopixe" as the directory for all runtime files. Disable setting !path when "geopixe" is open/closed to keep it off the !path.
 - g. Can close all projects, except "main".
 - h. How will this work in runtime session?
 - i. It should also be named "geopixe".
 - ii. And "geopixe" must be placed at same dir level as "main", so debugging from source can find runtime files.
 - iii. But a weakness to explicitly name the runtime dir as "geopixe".
- 2. GeoPIXE
 - a. database common
 - i. "startup" used to load database common (via "database_commons.def"). Move to "startupp".
 - b. Relocate "Test" code
 - i. Historical test routines in project "Test".
 - c. Build scripts
 - i. Open source form needs to use build scripts instead of Eclipse.
 - ii. Perhaps Eclipse "build" always runs "build-<project>" PRO in each <project>.
 - iii. Need to dodge around !path problem above.

- d. !path changes
 - i. find_device_objects
 - 1. Look in “geopixe” at level of “main”, using ‘file_dirname()’.
 - ii. startupp
 - 1. Now looks for “geopixe” runtime dir locally and one dir up.
 - 2. Also looks for “GeoPIXE.sav” locally, up and across to “geopixe” dir.
 - 3. Hence, should work with “geopixe” runtime dir and if dir gets renamed.
 - 4. Shows warning if no “geopixe” runtime dir found.
 - iii. Loading GeoPIXE.sav
 - 1. Background processes (Maia, DAQ, Back, Wizards, etc.) need to load GeoPIXE.sav without access to ‘startupp’ (as it’s IN GeoPIXE.sav).
 - 2. These must reference “geopixe” dir explicitly, relative to current dir.

11 Oct, 2022

Major discovery, using IDL 8.5.1 to start with, that “cd” is what causes the bogus multiple routine problems. Went through and removed all “cd”. Replace current dir with “file_expand_path('.')” or if a trailing slash is needed “fix_path(file_expand_path('.'))”. Also reworked all GeoPIXE.sav restores, which used “cd” in all Back plugins, Maia, DAQ, Blog, MM routines, Maia device, parallel worker, move_blog_image_plugin, Wizards and many “make” routines (e.g. string_packer).. All plugins (apart from Back, move_blog_image) are OK.

“cd” still in use in these routines, outside main GeoPIXE usage: ftp_connect, spawn_bridge_object (because “cd” is on the “other side”) and get_id, which s not used anymore anyway.

3. GeoPIXE

- a. startupp, find_device_objects, file_requester, image, geopixe_case_sensitive, geopixe_library, news, load_geopixe_image_stack, image_update_geopixe, retry_image_increment, sim_setup_geopixe:
 - i. Avoid using “cd”, replace using file_dirname() to pop dir or just “../”
 - ii. Leave “cd” in ftp_connect, spawn_bridge_object, get_id, move_blog_image_plugin.
- b. Version 7.6#

4. PIXE simulator

- a. missing (*pstate).p in calls to filter_update and detector_update.

5. Maia, Midas, UQ iXRF devices

- a. use ‘now = file_expand_path('.')’ instead of ‘cd,current=now’.

6. idl_query_geopixe

- a. replace ‘cd’ with file_expand_path('.')

7. Maia control, Scan List, DAQ Launch,

- a. Version 7.6#

8. Blog, Maia, DAQ processes

- a. Acitivity, Epics, DA2, ET_spectra, Group_spectra, Maia parameters and slow.

9. Wizards

- a. Depth, Fit_import (draft), Standards

10. parallel worker

- a. geopixe_parallel

11. geopixe_index

12. Browsers

- a. Maia, FalconX, MDADAQ2, Midas

13. plugins

- a. image move blog

14. string_packer

- a. make file

15. Back plugins

- a. Fix bad build setup in Eclipse

16 Sep, 2022

1. GeoPIXE

- a. Image_eventcb
 - i. Image_Process_Correct_Current
 1. Plot the correction versus X/Y.
 2. Rework extension of ends and interpolation.
 3. Add “filter” option for correction vector.
- b. Image_table_eventcb
 - i. OnCellSelect_Image_Table
 1. Take care of single row table case in forming ‘whole_column’.
- c. spec_evt

- i. If any 'q' pointer 'pq[]' is null/bad set it to 'ptr_new(-1)'.
- d. image_table_eventcb
 - i. image_table_evt
 - 1. set 'repeat_loop' = 1 if need multiple passes for large number of regions.
 - 2. Only 'write_spec' (again) if repeat_loop is set.
 - 3. Fix bug, which saw file not written. Go to version "@"
- e. translate_region
 - i. Return error if 'pimng' is not valid (no image displayed).
- f. spectrum_do_curve_fit
 - i. Tried to solve complexity around plotting spectra and fit curves, which are modelled at 0.1 channel spacing. The plots were scaled up x10, possibly preserving "area" or something like that.
 - ii. Solution is line @10-22, to scale down fit values x0.1.
- g. options_popup
 - i. Add "columns_check=n" to set the number of columns for check boxes.
- h. Version 7.6@

27 July, 2022

- 1. GeoPIXE
 - a. file_requester
 - i. Allow "+" in filenames. Enclose list of files in selection in "".
 - ii. Hide "+" during splitting of list.
 - iii. Strip off "" at end.
 - b. define
 - i. /spectrum defaults station=1
 - c. Fixed Image window layout (messy)
 - i. image
 - 1. Change initial Draw scr_xsize, scr_ysize to 360 (+trim).
 - 2. Set initial set state w,h = 360
 - 3. Change sizes for all variable widgets in X, new one for 'xsize_loglin'
 - ii. OnSize_Image
 - 1. Change min w to 376
 - iii. map_help
 - 1. Set threshold for help2 layout to >600 pixels
 - 2. Change xoff for wider widgets to 363.
 - d. evt
 - i. Adjusted layout for Linux
 - e. draw_spectrum
 - i. Set 'y2' to '(*pstate).yhigh' (from 'rescale_spectra').
 - ii. Do max with (*pstate).low + 0.01
 - f. Image_Process_Correct_Current
 - i. Now can select anchor point for normalization.
 - g. Image_Process_Correct_Current_file
 - i. Updates display afterwards.
 - h. open_vsub
 - i. req_vsub
 - 1. extract an error message from 'r.val.error'.
 - i. image_routines
 - i. analyze_image
 - 1. Accept 'i_uniform' element index to check element for uniformity.
 - 2. Use 'uniformity_mask()' to veto non-uniform pixels from 'q'
 - 3. Used by Wizard_standards.
 - ii. uniformity_mask()
 - 1. Use this to veto pixels that are not uniform (+/-20%).
 - 2. Called from 'analyze_image'.

3. Used by Wizard_standards.
- j. OnNotify_Image
 - i. Wizard 'sum-region' calls 'analyze_image' passing the 'uniform_element' and getting back some 'get_stats' for display in the results table.
- k. mode
 - i. new function to return statistical mode above a threshold
 - ii. default threshold is 1% of the mean
- l. spectrum_load_do
 - i. Do not need to form pp using /no_copy.
 - ii. Then do not need to make a new p from it afterwards, each p[] is done.
- m. spectrum_routines
 - i. Check y1,y2 range in plot and make sure Y2>y1.
- n. get_kvs
 - i. If key did not exist it returned no error, because 'exists_kvs' made error=0
 - ii. Should be error if key not found, now it does.
- o. version 7.6!
2. iXRF device
 - a. get_ems_ascii_spec
 - i. Follow EMS guide better and detect "Y" versus "XY" datatype.
3. plugins
 - a. Remove 'img_align' from all but 'image_align' plugin.
4. scan_list
 - a. Get "state" key from Excillum vsub in Timer 'coords-src-mode' loop, so that only set **(*pm).status.power** =1 when "state" is "on", as well as power above 95% of max.
 - b. In "start" set all **(*pj).raster.stroke = 0** before 'save_kvs_scan_list'.
 - c. Use "alarm_popup" if state not "on" while 'power_ignore'=0 and scan_sequence.active and **(*pstate).scan_sequence.raster_on** to show bad state alarm.
 - d. Set **(*pstate).scan_sequence.power_on=1** if power ON at start of sequence.
 - e. If sequence active, state is not 'on' and **(*pstate).scan_sequence.power** true then
 - i. pop-up a alert banner.
 - ii. If also **(*pstate).master** and state="ready" then push excillum state back to 'on'.
 - f. Set **(*pstate).scan_sequence.power_on=0** after a Stop.
 - g. version 7.6!
5. maia_launch
 - a. version 7.6!

23 June, 2022

1. GeoPIXE
 - a. open_kvs
 - i. Changed default timeout from 50s to 20s for all (e.g. 'get_kvs').
 - ii. Shorten the "lock up" when stale KVS accessed (for shapes).
 - b. spectrum_routines
 - i. draw_spectrum
 1. For 'show_diff' treat both Back1 and Back2 ("Back*") subtracted.
 - c. file_requester
 - i. Use 'file_search2' with /progress with "Find".
 - ii. In order for file_requester() to use the 'progress' bar it must use the combination modal=0 and no_block=0.
 1. If Modal=1 is used, it would force no_block=0 anyway, so this works the same, in effect. But this way (modal=0 and no_block=0) allows 'progress' to receive its events properly.
 - iii. Change the way 'Find' works. Uses 'find_path', 'pattern' and 'exclude' in common.
 - iv. Added the 'More' button to continue where previous Find left off.
 - v. Both call 'File_requester_find' now.
 - vi. Call 'file_search2' with all=0, so we just get first/next match.

- d. progress
 - i. New event processing that should not get triggered if modal=1 not used in parent.
- e. file_search2
 - i. 'exclude' brought to top level for "More" in 'file_requester'.
 - ii. Do not add the top level dir to 'exclude', so More can continue.
 - iii. Search in top dir last, then add the exclude.
 - iv. Add 'File' to p struct, so that progress can show the path.
- f. set_image_minmax
 - i. Add /reset_bounds option to redo bounds (e.g. if image rotated, etc.).
 - ii. **May invalidate a device with 'bounds.valid=1'.**
- g. Image_Process_Rotate
 - i. Call 'set_image_minmax' with /reset_bounds if a rotation, mirror, changes bounds.
- h. interelement_operations
 - i. Refine estimates, add better help prompts.
 - ii. version 7.6*
- 2. scan_list
 - a. mm_scan_list
 - i. Add option to 'mm_scan_list' for "per" or "mel" single arguments.
 - ii. version 7.6*
- 3. maia_launch
 - a. build_kandinsky_script_step_cal
 - i. Error in calculation of revised 'count' fixed (divide by 'iinc').
 - ii. Also limit 'iinc' to at least 1.
 - iii. version 7.6*

3 May, 2022

- 1. GeoPIXE
 - a. cut_evt
 - i. Add /All keyword for a single while spectrum CUT.
 - ii. Use energy range -3 to 100 keV.
 - iii. Take care: Type of "type" is uint.
 - b. evt_start
 - i. cater mode=5 (/all, whole spectrum CUT)
 - ii. Add extra 'file_ext' and 'file_title'.
 - c. evt
 - i. Add 'base_proj_file' to pstate
 - ii. Add "All Counts" to proj mode droplist
 - iii. Set map=0 for 'base_proj_file' for mode=4,5
 - iv. evt_proj_mode
 - 1. Set map=0 for 'base_proj_file' for mode=4,5
 - d. Export_images_csv
 - i. Needed to use "lxanes" for lines XANES case, not "xanes", which is for the XANES 3D images.
 - e. File Requester?
- 2. Image pileup plugin
 - a. Lost source. Retrieve from src folder in Release.
- 3. Image plugins w/ histograms
 - a. flux, raw_dead_fraction, raw_dwell_map, raw_flux
 - b. Make binsiz equal to 1% of maximum in image.

9 Mar, 2022

- 1. GeoPIXE
 - a. open_vsub
 - i. If a REQ endpoint is not provided, open the vsub without it.
 - ii. But still needs MMCL

- b. scan_dir_evt
 - i. Crashed for name="1234.dai", i.e. from "1234.dai.0", which should have been ignored.
 - ii. Now calls 'strip_file_ext' with /double to strip this extension too.
 - iii. And uses new 'strmid2' for vector arguments.
- c. strip_file_ext
 - i. Fails with vector arguments due to deficiencies in 'strmid'
 - ii. Use new 'strmid2' instead.
- d. strmid2
 - i. 'strmid' caused problems because strmid(s,start,len) would not accept 's' and 'len' vectors of same length. Write 'strmid2' to cope with this.
- e. version 7.6+
- 2. String routines
 - a. Strip_trail_zero
 - i. This was rolled back (to previous, 2015 version) due to a bug.
- 3. maia_control
 - a. read_maia_parameters
 - i. If filenames for Linear, Gain-trim, Pileup, Energy Cal do not contain a path separator (slash()), then prepend the 'path' for the parameters file itself.
 - ii. This way, can use local path, if all these files are kept together.
 - iii. This way, all files can be copied between systems and work, if kept in same dir.
 - b. write_maia_parameters
 - i. If the paths for Linear, Gain-trim, Pileup, Energy Cal are the same as the parameters file itself, then strip it off before writing it to the file.
 - ii. This way, can use local path, if all these files are kept together.
 - iii. This way, all files can be copied between systems and work, if kept in same dir.
 - c. maia_setup
 - i. Needed to use local struct *pl instead of *pm (and make a copy of *pm into *pl in case a tab change had not already done this).
 - ii. **Note that 'maia_setup_apply_hymod' actually assumes *pl already!**
 - iii. Read into *pl and copy back selected things (that aren't already done in 'maia_setup_apply_hymod': (*pl).file, (*pl).channel (latter a bit redundant because they'll be read back anyway).
 - d. version 7.6+
- 4. scan_list
 - a. Start adding code for Z probe
 - i. added comms5 everywhere
 - ii. added in retry_kvs
 - iii. added KVS entries (in mm-mel-1-cl KVS)
 - 1. Z probe endpoints:


```
{
    "SRC": "zprobe",
    "SUB": "tcp://mm-mel-1-cl.it.csiro.au:29510",
    "PUB": "tcp://*:29510",
  }
```
 - 2. config ('MM.Mel.SL.1.config') =


```
{
    PREFIX:
      STRING = 'MM.Mel.'
    ENDPOINTS:
      struct {
        STAGE:
          STRING = 'MM.Mel.ST.1.endpoints'
        SAFETY:
          STRING = 'MM.Mel.SS.endpoints'
        SOURCE:
          STRING = 'MM.Mel.EX.endpoints'
        KANDINSKI:
          STRING = 'MM.Mel.KA.1.endpoints'
        ZPROBE:
```

```
STRING = 'MM.Mel.LZ.1.endpoints'
```

```
}
```

```
}
```

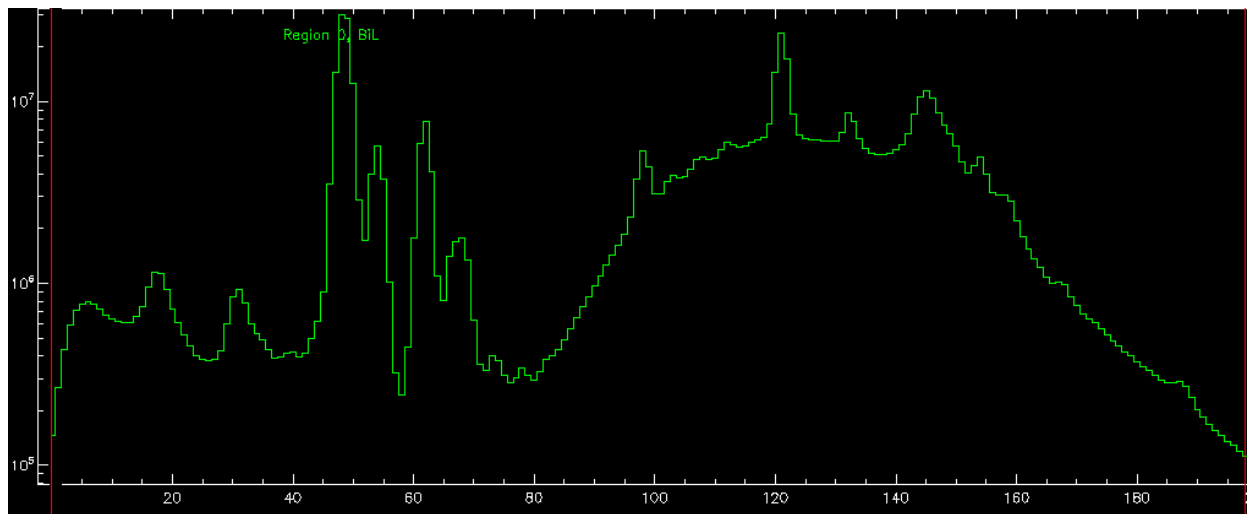
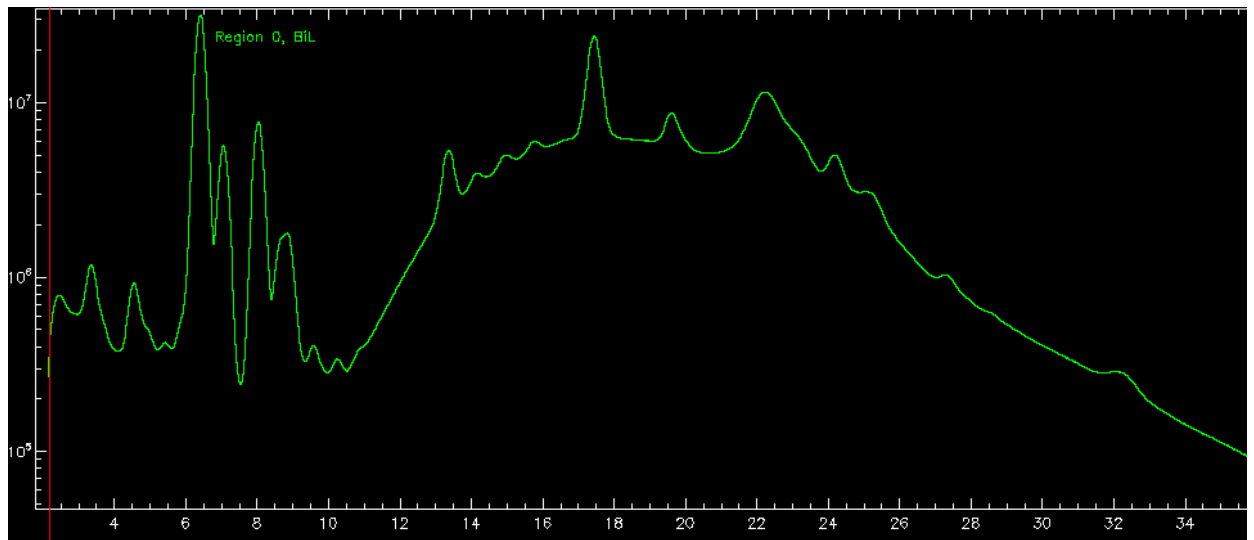
- iv. code added at init/creation, in retry and in state.
- b. Just grabs Z probe Z (mean) value for now, in “query” timer routine.
- c. version 7.6+

18 Nov, 2021

1. GeoPIXE

- a. Scan_dir_evt
 - i. Reject anything like “1234.dai.0” from blog file list.
 - ii. Skip .zarr dirs when scanning for blog files.
- b. Spawn_bridge_object
 - i. Detect error and return ‘error’.
- c. Pca_clusters
 - i. Make it display selected pixels coming from Corr or Image Region table better.
 - ii. Clusters display could still do with similar work ...
- d. Pca_cluster_routines
 - i. Analyze_pca
 - 1. Form ‘qref’ cross reference from whole image pixel index into subset pixel index ‘qs’
 - 2. Run ‘pcomp’ in /double mode.
 - ii. Notify
 - 1. ‘image-region-select’
 - a. Form ‘qpca’ (sub image index) from full image index ‘q’ in region, using the pca.qref cross-reference.
 - 2. pca_cluster_Analyze_Spline
 - a. Add ‘draw_pca_clusters’ to draw highlighted pixels.
 - 3. pca_cluster_select_Cluster
 - a. Add ‘draw_pca_clusters’ to draw highlighted pixels.
- e. Pca_cluster_eventcb
 - i. Postcreate ...
 - 1. Add ‘qref’ pointer to ‘pca’ struct.
- f. Corr
 - i. Add menu to clear selected pixels (qc).
 - ii. Make qc communicate between multiple Corr windows better, and back from region table. Also see selections coming from pca-clusters.
- g. Corr_eventcb
 - i. Corr_Analyze_Spline
 - 1. If highlight, then draw_corrs to show highlighted pixels.
 - ii. Corr_Clear_qc
 - 1. If highlight, then draw_corrs to show cleared highlight pixels.
 - iii. Notify
 - 1. ‘image-region-select’
 - a. If corr mode=1, then form ‘qc’ from region ‘q’.
- h. OnButtonCorr
 - i. Left mouse button: On highlight eq 0 do a notify ‘corr-analyze-clear’
 - ii. Right mouse button: On highlight eq 0 do a notify ‘corr-analyze-clear’
- i. plot_images
 - i. Fix to plot multiple selected element images, with centroids.
- j. evt
 - i. Now have extra ‘mode’ for projection droplist
 - ii. Change map for da_xanes_base2 (Energies), da_xanes_base1a (Collapse E), da_xanes_base1b (Proxy axis) bases to avoid mode=4 (evt_set_proj_mode, template, initial display)

- iii. Change sensitive for File widget to avoid mode=4 (evt_set_proj_mode, template, initial display)
 - k. cube_evt (NEW)
 - i. EVT sort without matrix into XYE data cube.
 - ii. Use compressed E scale, put back lookup in 'el' labels.
 - 1. $z1 = \text{uint}(\text{scale} * (\text{sqrt}(\text{energy}) - \text{sqrt}(\text{elow})))$
 - iii. accumulate using 'da_accumulate_cube'
 - iv. create image with type=2 (counts) and mode=4 (cube)
 - l. evt_start
 - i. Add 'cube_evt' to run for mode=4, xy_mode=0, sort_mode=0
 - m. image_table_eventcb
 - i. Plot numeric 'el' regions as log if not XANES.
 - n. image_history
 - i. to show XYE mode in details and compressed energies.
 - o. image_details
 - i. cater for mode = 4 for data cube with compressed energies axis
 - p. da_accumulate_cube (NEW)
 - i. wrapper for Fortran da_accumulate_cube
 - q. image_eventcb
 - i. image_export
 - 1. PNG appends to 'add' list like JPEG
 - r. string/file routines
 - i. strip_file_m, extract_extension
 - 1. Cater for paths with embedded "."
 - s. periodic_table
 - i. Enabled Ra, Ac.
 - t. evt
 - i. evt_set_evt_file:
 - 1. If set first blog file, then clear last field (evt_set_evt2_file).
 - u. Version 7.6~
- 2. Fortran
 - a. da_accumulate_cube (NEW)
 - i. accumulate an XYE datacube using compressed E (supplied).
- 3. Maia Control
 - a. Maia_launch
 - i. Detect error return from spawn maia, blig clients and destroy obj and exit.
 - b. spawn_maia_clients
 - i. If any spawn_bridge_object call fails, exit with error=1
 - c. spawn_blog_clients
 - i. If any spawn_bridge_object call fails, exit with error=1
 - d. maia_setup
 - i. maia_setup_apply_groups
 - 1. Added setting: "activity.select.pileup" and "activity.select.throttle"
 - e. Version 7.6~
- 4. CHESS HDF5 device
 - a. Hack of NSLS HDF device to adapt to old CHESS data from F3 for Jess.
 - b. Header read from text file puts chess_flux array in common.
- 5. Compress spectrum plugin
 - a. Map 2-35 keV as a SQRT() onto 0-199 for a compressed spectrum for full data cube.



6. Pulser FWHM map spectrum plugin
 - a. Now transform to X channel for each spectrum, in case (first) one uses a very different E cal.
 - b. Seems to work for (very) mixed E cal and no cal (cal A=1, B=0).

28 Oct, 2021

1. GeoPIXE
 - a. Read_tif_images (NEW)
 - b. Geopixe_defaults
 - i. Add 'custom' fields (enable, lab, path, file).
 - c. Read_geopixe_images
 - i. Had become corrupted? Copy back from geopixe2.
 - d. Image_Table_eventcb
 - i. OnButton_Image_Table_Export_XFM
 1. Export region Box extents to custom path/file.
 - e. Image_routines
 - i. Plot_curve (and all plot shapes)
 1. Thick must be scaled by xscale for curve and centre-line.
 - ii. Plot_handle
 1. Set scale to 4*thick, if thick not equal to 1 (i.e. for plot case)
 2. Plot handle only for distance and S.pixel.
 - f. Image_eventcb
 - i. Image_process_plugin
 1. Make history consistent with code in Notify routine ("Plugin ...")
 - ii. OnNotify_Image
 1. Use case on ops.name now.
 - iii. Image_save_all_tiff

1. If file name is fnumeric, then set 4 places.
 - g. Image_process
 - i. Add “* Correct zero pixels” to list.
 - ii. Make all that set multiple planes have a “*” at the start, like with Plugins.
 - iii. Make sure same/consistent text is added as a history record.
 1. Fix all image_process_* routines ...
 - h. Image_process_eventcb
 - i. OnButton_Image_Process_Get
 1. Make ‘List’ all text up to either “[“ for tag element, or a separator [,:()) – note no whitespace used as a separator to accept multiple word methods.
 2. For “process” case, check tag element field at end (e.g. “[Ni]”) against current element ‘i’ and skip if no match. This ensure a “*” method gets done only once (for all).
 3. Strip off the [tag] before test against ‘list’.
 - ii. Operation order
 1. Need to think about whether we need to introduce a "priority" in the order of these operations. For example, should we do a global "Correct Zero pixels" before individual median filters? ; Or should that be the other way round? Should all operations performed on an image be tagged by a sequence number, so we can do them again in the same order?
 - i. Version 7.6x
2. Notes on image processing
 - a. List
 - i. Names need to be unique and useful. Currently, stopping at first space, so only have “removed” for remove zeroes, which is not great.
 - ii. Need to make Image Process **List** consistent with **History** that is added.
 - b. ‘ops’ struct for ‘image-process’ Notify (populate in ‘OnButton_Image_Process_Get’)
 - i. Mode: 1 (pass # image plane), 0 (use current image)
 - ii. Image: image plane to act on
 - iii. Name: name of operation
 - iv. Valid: 1 (not used?)
 - v. Operation: routine name
 - vi. Arg1: Numeric (e.g. filter width, erode value)
 - vii. Filter: filter name, only used in ‘inter-element’
 - viii. Arg2: string
 - c. modes
 - i. name = plugin
 1. Use: image, name, operation
 - ii. name = inter-element
 1. Use: image, name, operation, arg1, filter, arg2
 - iii. name = process (e.g. menus and Process list)
 1. Use: image, name, operation, arg1
 - d. Doing the processing in Image Notify routine
 - i. Name = “plugin”
 1. Handle a plugin
 - a. Uses only ‘operation’ and image index
 - b. Image index = ‘image’ (if ‘mode’=1), else (*pstate).image
 - ii. Name = “Inter-element”
 1. Handle inter-element operations
 - a. Uses ‘operation’, ‘arg1’, ‘filter’, ‘arg2’
 - b. Operation = inter-element operation (e.g. subtract)
 - c. Arg2 = source element
 - d. Arg1 = weight
 - e. Filter = pre-filter on source
 - iii. Name = “process”

1. Handle image processing
 - a. Uses only 'operation', 'arg1' and image index
 - b. Image index = 'image' (if 'mode'=1), else (*pstate).image
- e. "Get" button to replay processing
 - i. What about operations that implicitly do ALL element planes?
 1. Plugins with "*"
 2. Things like "Correct Zero pixels"
 3. Can't just repeat for every element that shows the history.
 4. Clear border OK, as it uses select for each element.
 - ii. What if they need a target element to use (e.g. "Correct Zero pixels")?
 1. Appended as " [Ni]"
 - iii. Solution:
 1. Look for the element appended and test against target element.
 2. Only do process if there is a match (only gets done once).
3. String routines
 - a. Strip_trail_zero
 - i. Allow places to not only set a maximum number of places, but a minimum too.
4. Fortran
 - a. Da_accumulate11, 'da_accumulate10' and 'maia_accumulate_dtfx2' and 'maia_accumulate_dtfx2_3D'
 - i. Accommodate (unsigned) X,Y larger than 32k, using local long 'xl','yl'.
 - b. Version LIB 53
5. Maia Control
 - a. Misguided attempt to have ROI rates.
 - i. Not real-time enough.
 - ii. Better to use pileup limits to control and effective ROI rate, which gives real-time updates.
 - b. Maia_rates
 - i. Comment out the misguided ROI rate fields.
 - ii. Using pileup to control an effective ROI rate from Hymod works much better.

29 Sep, 2021

1. Spring8 U37 device
 - a. Mod to enable it to read a truncated input TXT file.
2. GeoPIXE
 - a. Read_png_images (NEW)
 - b. Obj_new:
 - i. Fix reference to "new_obj()" instead of 'obj_new()' in 'read_pnf_images', 'read_png_images', 'read_bmp_images'.
 - c. Image_preview
 - i. Give starting values for R,G,B in case they don't get set.
 - d. Image_import (image_eventcb)
 - i. Add case for type= 3 for PNG.
 - e. Image
 - i. Add menu and call 'image_import' for type=3.
 - f. Round_kernel
 - i. Convert input INT to long to ensure can have a large kernel.
 - g. read_pnc_images for APS device
 - i. Fixed the same obj_new bug in 'read_pnc_images'.
 - h. Define
 - i. Add ROI.rate to 'maia_struct' for ROI count-rate
 - i. Export_images_csv
 - i. Treat the case of XY mode, non XANES separately, so we write separate files per region with element columns.
 - ii. XANES XY still produces a separate file per element and with region columns.

- j. Image_table_eventcb
 - i. Image_table_EVT
 - 1. If /by_detector, then do not use multiple passes, as 'sort_evt' will only use one (detector_select) mask.
- 3. Standards wizard
 - a. 'wizard-return' event
 - i. If error, pop up warning. "Cancel" to abort all, else continue.
 - b. Version 7.6v
- 4. Maia Control
 - a. Maia_scratch.def
 - i. scratch_ROI_rate = 22
 - b. Maia_launch_initial
 - i. Set scratch.datum[scratch_ROI_rate].key = "Maia:ROI.rate"
 - c. Maia_launch
 - d. Maia_update_et_spectra
 - i. Set (*pl)[2:3] to ROI limits, enable using (*pl)[2]
 - ii. Get fraction back through (*pf)[2] and set new (*pm).ROI.rate
 - e. Maia_update_parameters
 - i. Set scratch in Kandinski for ROI to (*pm).ROI.rate
- 5. Maia_rates
 - a. Add a ROI rate below flux1 rate
 - b. In 'maia_rates_highlight', set ROI rate text to (*pm).ROI.rate
- 6. Blog_client_ET2_spectra
 - a. Use (*pl) 2,3,4 for ROI enable, low, high
 - b. Fraction is total 'e' within (low,high) / total events 'n'.
 - c. Uses (*pf)[2] for the fraction returned.

6 Sep, 2021

- 1. GeoPIXE
 - a. Evt_start
 - i. Call 'check_mpdam' to check MPDAM file, Correct, all DA matrices, phases.
 - ii. Use returned 'mpdam' = 'file' to set (*p).file[] and widget in EVT.
 - iii. NOTE: Side track to fill in Normal DA and phase maps, only works in Cluster=1 mode at present.
 - iv. Also provide 'geopixe_do_DA' side-track in non-cluster mode. However, it will not work in IDI VM mode.
 - v. Put call to check_mpdam outside "verify" lines.
 - vi. Pass verify to check_mpdam.
 - b. Check_mpdam (NEW)
 - i. Tracks down all files: correct file, all referenced DAM files.
 - ii. Checks that phase maps data source match raw data source. If they don't, post a warning that they'll now be generated.
 - iii. Stringifies mpdam struct and passes back, if arg_present().
 - iv. New arg "verify", else silent mode.
 - v. Pass silent to read_mpdam.
 - c. Write_mpdam (NEW)
 - i. Write new MPDAM file
 - d. Evt
 - i. Write mpdam using 'write_mpdam'
 - ii. If 'mode=3' append "-MPDA" to DAI filename (a few places).
 - iii. Do not use 'build_output_path' in Notify 'start-evt'. Just use extract_path() on the supplied output filename.
 - e. Cluster_merge_images
 - i. If images=pp arg not present, then free_images, pp.
 - f. Geopixe_do_commands

- i. Detect MPDA and test if phases good using 'check_mpdam'.
 - ii. If not, do normal DA if not found.
 - iii. Check loaded normal DA file for mode=0, type=0, correct=0.
 - iv. Correct yields and project phases.
 - v. Then continue with MPDA.
 - vi. Tests ARGV for blank entries from command line ""
 - g. Correct_yield
 - i. Now calls 'correct_yields_do_correct', 'correct_yields_do_phases'
 - h. correct_yields_do_correct (NEW)
 - i. Does the yield correction, with option for /progress display.
 - i. correct_yields_do_phases (NEW)
 - i. Does the phase projection,
 - j. Cluster_merge_images
 - i. Free 'pp' images if not passed back.
 - k. Geopixe_do_da
 - i. Fix "error" versus "err".
 - ii. Return error=1 if cancel from normal DA.
 - iii. Return 'mpdam' file-name, if it changes.
 - l. Plot_images
 - i. Fix bug that stopped JPEG mode from doing multiple elements.
 - m. Parallel_client
 - i. Parallel_init
 - 1. Use progress bar as spawn background processes.
 - n. Cluster_client
 - i. Call 'parallel_init, /progress'
 - o. Warning_popup
 - i. Need to save group in pstate for close later, if local.
 - p. Read_maia (actually not in Maia device dir)
 - i. If AA, BB only found deep in file, then just goto, done (;@9-21)
 - q. Scan_dir_evt
 - i. Aim for output file name with tags (-cuts, -MPDA) based on 'mode' and look for these first.
 - ii. If not found use normal DA output file.
 - r. Batch sort:
 - i. batch_correct_output (on Notify of EVT output changed)
 - 1. correct output file name based on 'mode' to include tags (-cuts, -MPDA).
 - s. Version 7.6v
2. 'bin' under Linux
 - a. geopixe
 - i. Needed to enclose args in double quotes:


```
exec $idl -rt=$sav -args "$1" "$2" "$3"
```
 - b. But this creates the problem of passing null args into argv.
 - i. Need to filter these out in 'geopixe_do_command'

18 Aug, 2021

1. IDL Warning
 - a. MPA4 device was crashing. Fixed it by deleting the **fastcom_mpa4_device__define.OLD** file from the project. It should not compile this, so why does this matter?
 - b. **DO NOT KEEP .old files in projects!** Move them to the 'old' dir.
 - c. **This may have been a red herring**, as an X,Y out of range error was fixed later.
2. GeoPIXE
 - a. Evt_start, Da_evt
 - i. Add test to check that phase maps have been created from this input raw data-set.
 - b. Evt
 - i. Make "Energy proxy axis" and "Energies" not visible for MPDA mode.

- ii. Do for initial display and when projection mode changes.
 - iii. Calls 'define_devices' with new arguments to get all device names.
 - iv. Now uses 'instance_device_objects' to create an objarr for all devices.
 - c. Fit_setup
 - i. Just layout tweaks.
 - d. Str_break
 - i. Return broken up string (w/ double quotes intact) in NEW 'all=all'.
 - e. Correct_yield
 - i. When image changes, Notify 'original DA matrix' and update Original DA.
 - ii. Now a "Load" of a new .correct file will still keep using 'original' DA matrix from attached images.
 - f. Define
 - i. Add 'top' to define(/ wizard_notify).
 - g. Wizard_check_window_id (NEW)
 - i. Checks event.top ID against 'windows_open' to detect multiples open,
 - ii. Keeps list free of invalid IDs and duplicate IDs.
 - iii. Returns a 'count'.
 - iv. Use in any Wizard in 'wizard-return' processing to check for multiple windows.
 - h. Wizard_check_windows (NEW)
 - i. Check that all windows 'windows_needed' are open.
 - ii. Use in any Wizard before starting an operation that talks to 'windows_needed'.
 - iii. Stop using 'windows_last'.
 - i. Wizard_window_count (NEW)
 - i. Returns a count of windows of a given type.
 - j. Wizard_resize
 - i. Resize a tree of widgets, managed by flags in the 'uvalue' for each widget (including base widgets). Use anywhere a specified size is given initially.
 - 1. uvalue = {xresize:1, yresize:1, help:'normal context/tracking help text'}
 - ii. Can be used for non-Wizard programs too.
 - iii. Pass in the current XY size of the TLB and the minimum size for the TLB.
 - iv. Remember to use 'widget_info(event.top, /geometry)' to update TLB size after the resize has finished.
 - k. Notify 'wizard-return'
 - i. In routines: evt, fit_setup, OnNotify_image, OnNotify_image_table, OnNotify_imageRGB, OnNotify_spectrum, OnNotify_select
 - ii. Add 'pwiz' pointer to pstate in each of these for Wizard return Notify.
 - iii. Sets 'top' in return struct to local 'event.top'.
 - l. Make_corr_tvb
 - i. Need to use 'el_code' not 'atomic_number' for ex,ey to handle L shell (e.g. 'PtL'), to detect true elements, that will need norm to charge map.
 - ii. Fixed this bug created on July 19, 2021 (version 8.6p).
 - m. Spec_evt
 - i. Fixed odd mix of mean(live) and mean(1/live) – see ;@9-21
 - 1. This seems to sit OK with pixe_fit, which does use 'dt_corr' factor.
 - 2. pixe_fit seems to save charge (assumed to be 'live') after correcting for 'dt_corr'.
 - ii. Now call new obj methods to smooth dead_fraction and flux maps.
 - n. Define_devices
 - i. Now returns vectors for current title, names of Device objects (not generic).
 - o. Instance_device_objects (NEW)
 - i. Returns an Obj array of all device names supplied.
 - p. Base device
 - i. Add dummy methods for smooth - weight, flux.
 - q. Version 7.6s
3. Geopixe_update

- a. ftp_update
 - i. ftp_defaults
 - 1. Use all=s2 in call to 'str_break' to keep single " in a password.
- 4. Wizard_standards
 - a. Notify 'wizard-return'
 - i. Now keeps count of 'open-test' returns to test for duplicate windows open.
 - ii. If duplicate a warning is popped up to suggest closing duplicates.
 - iii. Uses 'Wizard_check_window_id' to maintain a list of valid open IDs.
 - iv. After Warning pop-up, set a count-down so that we only test for windows after a delay of a few cycles (about 32s – increased timer to 8s).
 - v. This also deals with IDLDE issues with delayed events, which can cause overruns, which leads to clusters of events that give false window counts.
 - vi. Also tests for 'wizard-return' from other Wizards, and pops up a warning.
 - b. Stop using 'windows_last'.
 - c. Test TLB valid before test for window or event routine processing.
 - d. Add "figure" button to re-display the figure for this tab.
 - e. Use new 'wizard_resize'.
 - f. Free more pointer heap in cleanup.
 - g. Fix bug in plot versus line energy.
 - h. Version 7.6s
- 5. Wizard_depth
 - a. Notify 'wizard-return'
 - i. Now keeps count of 'open-test' returns to test for duplicate windows open.
 - ii. If duplicate a warning is popped up to suggest closing duplicates.
 - iii. Uses 'Wizard_check_window_id' to maintain a list of valid open IDs.
 - iv. After Warning pop-up, set a count-down so that we only test for windows after a delay of a few cycles (about 32s – increased timer to 8s).
 - v. This also deals with IDLDE issues with delayed events, which can cause overruns, which leads to clusters of events that give false window counts.
 - vi. Also tests for 'wizard-return' from other Wizards, and pops up a warning.
 - b. Stop using 'windows_last'.
 - c. Test TLB valid before test for window or event routine processing.
 - d. Add "figure" button to re-display the figure for this tab.
 - e. Use new 'wizard_resize'.
 - f. Free more pointer heap in cleanup.
 - g. Redo plots when resized.
 - h. Fix bug in Clear table.
 - i. Enabled "Verify" in "Sort-Image" Notify, to enable searching for file-name matches for first image sort.
 - j. Version 7.6s
- 6. Wizard_import_fit (still a DRAFT only)
 - a. Notify 'wizard-return'
 - i. Now keeps count of 'open-test' returns to test for duplicate windows open.
 - ii. If duplicate a warning is popped up to suggest closing duplicates.
 - iii. Uses 'Wizard_check_window_id' to maintain a list of valid open IDs.
 - iv. After Warning pop-up, set a count-down so that we only test for windows after a delay of a few cycles (about 32s – increased timer to 8s).
 - v. This also deals with IDLDE issues with delayed events, which can cause overruns, which leads to clusters of events that give false window counts.
 - vi. Also tests for 'wizard-return' from other Wizards, and pops up a warning.
 - b. Stop using 'windows_last'.
 - c. Test TLB valid before test for window or event routine processing.
 - d. Add "figure" button to re-display the figure for this tab.
 - e. Use new 'wizard_resize'.
 - f. Free more pointer heap in cleanup.

- g. Version 7.6s
- 7. Flux, Raw flux image plugins
 - a. Now plots a histogram of flux per pixel.
- 8. Fastcom MPA4 device
 - a. Follow through from busy, veto and weight to dead_fraction and dwell maps.
 - a. Add methods for 'get_dead_weight', 'get_dead_weight_mode' and 'get_dwell'.
 - b. Add methods for smooth weight, flux. Use these implicitly for get_dwell and get_weight.
 - c. Now uses 'mpa4_events2' for events, which now uses 'channel_on'.
 - d. Add routine for 'mpa4_accumulate_dtfx' to accumulate dead-fraction and flux.
 - e. Fortran has a bug in Lib version 51 and less.
 - f. Fixed a bug in 'mpa4_events2' passing 'n_fx' as INT instead of Long.
- 9. Fortran
 - a. Da_accumulate10
 - i. Tried not norm to 'sum', as it should be 1.0 anyway, normally.
 - ii. Only helped 2%, so revert back.
 - b. Mpa4_events2
 - i. At line 80, use test on (k.ge.0) not on (adc(k).gt.0) as before.
 - ii. This is best, as k (=ADCindex(i)) is negative for no event.
 - iii. Add 'charge_adc' and 'channel_on' selection. Return events for all ADCs in channel_on plus the charge_adc and all channels for busy/count.
 - c. mpa4_accumulate_dtfx
 - i. Accumulate 'dead_fraction' and 'maia_weight' (which is also dwell in ms).
 - ii. Accumulate charge pulses from 'charge_adc' in pseudo stream.
 - iii. Crash bug was caused by having flux(x,y) increment OUTSIDE of the test for X,Y bounds.
 - d. Version 52

28 July, 2021

- 1. GeoPIXE
 - a. Export_images_csv
 - i. /plot option to plot XANES spectrum for current region.
 - b. Image notify: 'image-region-select'
 - i. If XANES=0 and an energy proxy axis used, then plot current region as XANES spectrum, by calling 'export_images_csv' with /plot.
 - c. Inter element operations
 - i. Revise Auto to use a histogram 1% cutoff on the ratio to select factor 'f'.
 - ii. Leave median(3) on both source (after filter) and target.
 - iii. Only consider pixels with source above 50% max and target above 5% max.
 - d. Geopixe_defaults
 - i. Added shape.enable, shape.file, shape.path
 - ii. To denote saving current Box shape to this path/file.
 - e. Image, image_eventcb
 - i. Form 'shape_file' from prefs.shape
 - ii. Added 'shape_file' to pstate, passed via 'Postcreate_image_base'
 - f. Set_kvs_box (image_routines) called from 'onbutton_image'
 - i. Write a file given by '(*pstate).shape_file, if provided.
 - ii. First test if path needs creating using 'safe_file_mkdir'
 - iii. Save X,Y origin on first line, X,Y size (mm) on second.
 - g. Fit_setup
 - i. Realize detector mode – set sensitive to 'yields.array'.
 - ii. Detector mode – because we do a detector_update to reset list, need to also set_combobox_select.
 - iii. Add 'file' to pstate to store file-name for XANS energies or MPDA master weights spectrum.

- iv. Change Generate DA matrix pop-up to 3 modes and solicit file-name for master weights here now.
- h. Calc_da_matrix2
 - i. Now get master weights file in fit_setup, not here
- i. Spectrum_display_eventcb
 - i. Spectrum_process_plugin
 - 1. Copy back 'p' array (i.e. '(*pstate).p) = p) in case plugin has changed the array, or reduced the number (e.g. combined them).
 - 2. Initialize a new '(*pstate).pshow' too.
 - j. Version 7.6r
- 2. Fastcom MPA4 device
 - a. Added busy, veto and weight.
- 3. MPDA Master Weights spectrum plugin
 - a. Form a weighting spectrum from all (region) spectra passed to it.
 - b. Returns just a single spectrum 'p', which needs a mod in 'Spectrum_process_plugin'
- 4. Fortran:
 - a. MPA4_event
 - i. Added busy, veto and weight.

23 July, 2021

- 1. Fastcom MPA4 device
 - a. Only supports: time_patch=5b
- 2. Fortran:
 - a. Fixed byte order and masking to prevent sign-extension (after OR) trashing top bits.
 - b. Version 46

19 July, 2021

- 1. GeoPIXE
 - a. Sum_peaks
 - i. Found that if it merged peaks within 60 eV (in histogram analysis), it must not do this for 'Comptons'. Instead, it will merge all non-Compton peaks and leave Compton sums as individual lines in sum peak, but still labelled as 'Compton' or 'Compton2'.
 - ii. This ensures they still get Compton treatment in 'line' routine, which broadens them.
 - b. Fit_setup
 - i. Load_pcm_parameters
 - 1. Use file_requester(/skip_if_exists) approach to determine the file path for detector layout file, before calling 'read_detector_layout'.
 - 2. Also in onrealize routine.
 - 3. Copy back new filename to detector struct in both cases.
 - c. Interelement_operations
 - i. If close window, set operation to 'off', to cancel the effect, and send Notify to show it cancelled.
 - ii. Do NOT skip update, if operation changed to 'off'. Must still Notify.
 - d. Corr_routines
 - i. Make_corr_tvb
 - 1. Also check if axis element (cx, cy as ex) has a valid atmic_number(). If not, do not normalize to charge map.
 - e. Read_event_buffer
 - i. Add ascii option to read formatted Hex and convert using a 'reads', as used in MPA4 device.
 - f. Source_update (NEW)
 - i. Reads all source definitions/savs, like filter_update.
 - ii. Not used yet: should get used where there is a source droplist, etc.
 - g. Hex_jam
 - i. Convert the Hex values in string (array) 'str' to an integer value (array).

- ii. Assume they are padded to correct byte length.
- h. Version 7.6p
- 2. Falstcom MPA4 device
 - a. Added a driver for MPA4, for Madrid data
 - b. Only supports: time_patch=5b
- 3. Atlas iXRF device
 - a. Added import of EMS spectra files.
- 4. Test:
 - a. GeoPIXE/device/MPA4
 - i. Test_mpa4
- 5. Fortran:
 - a. Added 'mpa4_events' routine for MPA4 Fastcom data from Madrid (time_patch=5b).
 - b. Version 45

13 July, 2021

- 1. GeoPIXE
 - a. Find_file2
 - i. Use /strict for all 'inumeric' and 'sort_file_numeric' calls.
 - ii. This helps clarify returns used to file_requester for dir lists/ trees.
 - b. Long64test, sort_file_numeric
 - i. Add /strict keyword, pass to 'inumeric'
 - c. Scan_dir_evt
 - i. Save (in 's0') raw file-names to use if no analysis DAI files found.
 - ii. But remember to strip file extensions.
 - d. File_requester
 - i. When a node of the tree is selected, using /set_tree_visible too, to set scroll view in range:
 - 1. widget_control, id, /set_tree_select, /set_tree_expanded, /set_tree_visible
 - ii. except when it is selected/clicked on (as it must be visible already).
 - e. Image
 - i. If 3D stack (xanes mode), use menu "Energy Associations" and set title to Corr to "Energy Associations" too.
 - f. Corr
 - i. Fix title if showing energy Associations in XANES mode.
 - g. Evt_start
 - i. Remove debug (/debug) from Throttle file_requester call.
 - h. Version 7.6n
- 2. Spring8 37U device
 - a. Notes
 - i. Reads single channel data, and normalizes to Flux0 (I0).
 - ii. Cannot make use of ICR, as there is no time stamps or dwell data.
 - b. Routines
 - i. Spring8_37U_device__define
 - ii. Get_spring8: loads .mdaq header data.
 - iii. Analyze_spring8:
 - 1. Reads data files and creates header file .mdaq
 - 2. Create I0 and ICR map binary files.
 - iv. Spring8_flux0: Read I0 map
 - v. Spring8_ICR: Read ICR map.
- 3. Bin
 - a. Remove "&" on exec line in scripts for geopixe and mm_scan_list, so they are not detached. This keeps terminal window open under Linux with new security feature to timeout open terminals.

9 July, 2021

1. GeoPIXE
 - a. Batch_sort
 - i. Scan_dir_evt
 1. 'output' for 'build_output_path()' in final loop must be the full (raw) path, else proot gets reset.
 - b. Write_geopixe_zarr
 - i. Fixed bug in dir name, which adds extra /.zarr/ dir level.

8 July, 2021

1. Maia_control
 - a. Version 8.6m

6 July, 2021

1. GeoPIXE
 - a. write_geopixe_zarr
 - i. Get this error (laptop). But "Resume" continues to run OK ...
 - ii. % PYTHON_CALLMETHOD: Exception: [WinError 5] Access is denied: 'C:\\Software\\Demo\\MM\\analysis\\924\\924-x.zarr\\.zarr\\pcal\\0.icvtj3z0.partial'
 - iii. -> 'C:\\Software\\Demo\\MM\\analysis\\924\\924-x.zarr\\.zarr\\pcal/0'.
 - iv. % Execution halted at: PYTHON::_OVERLOADMETHOD 658 python__define.pro
 - v. % WRITE_GEOPIXE_ZARR 197 write_geopixe_zarr.pro
 - b. Image_Export_Zarr_GCF
 - i. Create a GeoPIXE Command file (.gcf) to export current image as ZARR.
 - c. Image
 - i. Add menu to create a GeoPIXE Command File to export image as ZARR.
 - ii. Geopixe entry point also does 'startupp,/database,/colours,/devices' or alternatively, it could call 'image,/init_only'.
 - d. export_zarr (NEW)
 - i. Wrapper to read an image file and call 'write_geopixe_zarr'.
 - e. geopixe_do_commands
 - i. To cater for input "files=" that are not raw data files, on header read error, skip the header parameters.
 - ii. Now supports export commands ('export_zarr').
 - iii. For regions, need to write back cut-down 'evt_files' list into 'str[files_index]' and re-join str to form 'args' for each pass.
 - iv. Put a limit of 3 files on cluster mode (<4 and it reverts to cluster=0 mode).
 - f. Evt_start
 - i. Put a limit of 3 files on cluster mode (<4 and it reverts to cluster=0 mode)..
 - g. File_requester
 - i. If /write, then make 'path' equal to 'file' path, if it exists.
 - h. Open_comms
 - i. Default server → mm-mel-1-cl.
 - i. Crash issue
 - i. If run from mm-mel-1-cl (which cannot connect to external servers) with KVS enabled and a Perth KVS specified, GeoPIXE will crash out.
 - ii. The IDL-python bridge does not handle the "No response from server" error properly and crashes.
 - j. Version 7.6m
2. Install and maia_update
 - a. Changes to install.pro require BOTH these to be compiled.
3. Idl_query
 - a. Added 'python_version()' to output, with error catch.

29 June, 2021

1. GeoPIXE
 - a. geopixe_do_commands
 - i. Pick up the following from input data header: sample, grain, comment, charge, IC parameters, dwell.
 - ii. Could treat a change in scan range (see code), but this would entail passing stringify(p regions) instead of 'mfile'. But this string can get extremely large. So instead, just detect an incompatible scan.
 - b. Vector_merge
 - i. Treat arrays of pointers or objects explicitly, else they becomes Lists.
 - c. Export_images_csv
 - i. Add 'collapse_done' flag, and do collapse of ProjectX,Y shapes, if collapse_done=0.
 - ii. Add ProjectX,Y default collapse to regions section of code.
 - iii. Fix row titles.
 - iv. Add major_x and major_y flags (default to major_y) for out ordering.
 - v. Correct output generation if no X,Y cords labels provided.
 - d. Open_kvs
 - i. Added /List support to 'set_kvs'. This uses a loop and lpush method, so is a bit slow.
2. String routines
 - a. Unstringify
 - i. Allow a simple string to 'unstringify', which will now be simply returned as is, unless the first character has significance (i.e. {, [, ", ', P, Complex, Double, or O), or it is detected to be a valid gnumeric(), which then gets returned as a number.
3. Scan_list
 - a. Swap new code for linear encoder and ERRNO codes to source3, and return older code to source2.
 - b. Use source2 in Perth (w/ python 2.7 and IDL 8.5.1).
 - c. Use source3 in Mel (w/ python 3.8 and IDL 8.8).
 - d. Version 7.6k
4. Install and maia_update
 - a. Changes to install.pro require BOTH these to be compiled.

25 May, 2021

1. GeoPIXE
 - a. Define
 - i. Add 'energy_proxy_axis' for 2D proxy axis to define energy index
 1. 0=none, 1=X, 2=Y
 2. Axis gives index into energies list in 'energies_file'.
 - ii. Always read 'energies_file' now in 2D (xanes=0) and 3D (xanes=1) modes.
 - b. Write_geopixe_image
 - i. Save 'energy_proxy_axis' and 'energies_file', if xanes = 0 (2D image)
 - ii. Always write 'energies_file' now in 2D (xanes=0) and 3D (xanes=1) modes.
 - iii. Version = -57
 - c. read_geopixe_image
 - i. Seemed to not set 'original_zsize' correctly, before version 54. Now fixed.
 - ii. Read 'energy_proxy_axis' and 'energies_file', if xanes = 0 (2D image)
 - iii. Version = -57
 - d. Evt
 - i. Add a "Energy proxy axis" droplist to the DA/E.Cal tab, to select between "None", "X axis" and "Y axis". If not None, then show Energies widgets.
 - ii. Resuse the energies widgets from 3D modes.
 - iii. Single old map base for energies, now split into 3 map bases for: proxy axis, energies and the 3D Collapse energy and XANES element ID.
 - iv. Add "C*" button to generate a GeoPIXE Command File.
 - e. Da_evt

- i. Read_da without e_beam=0.0 to read all DA matrices in Line XANES mode.
 - ii. 'z_coords' set to eDA[e_lookup] as energies for proxy axis pixels.
 - iii. If line_xanes then Set px_coords or py_coords to z_coords, according to proxy_axis.
 - iv. Note: in cluster mode the proxy axis index must be the absolute axis value, not just the value in this stripe. Need to add the offset3 before accessing e_lookup[].
 - v. Also, veto compress on proxy axis, in Line_Xanes mode.
 - vi. **Tweak code in yrange3 in cluster mode.**
- f. Calc_da_loop
 - i. Also test for 'scatter' to move peaks as Ebeam moves in XANES.
 - ii. Also merge the main matrix, as done for pdm.
- g. Read_da
 - i. Only return something in eDA, if extra DA matrices appended.
- h. Da_accumulate11
 - i. Like 'da_accumulate5', but with multiple energy planes in 'matrix'.
 - ii. Pass in 'z' vector of energy plane index and 'n_energy' of third dimension..
- i. Export_images_csv
 - i. In regions mode, if shape mode "ProjectX" or "ProjectY" used, then collapse the region onto the lowest XY and average across collapse direction. These then report as 1D traverses.
 - ii. In XY mode, sort output on XY so that any axis with p?_coords found varies fastest, so that each value of the other axis is grouped together and the fastest ones vary within that. But don't do this if multiple regions are tabulated.
 - iii. Non XY mode still outputs full rectangle, zeroing pixels outside shape.
 - iv. Add the (*pstate).region_id to the filename.
- j. Evt_start_image_increment2
 - i. Need to also accumulate all py_coords values as we go.
 - ii. Removed 'obj' argument, not used (old device index).
- k. Retry_image_increment
 - i. Remove 'obj' argument to 'evt_start_image_increment2'.
- l. Options_popu
 - i. Set check box column=2 now.
- m. Image
 - i. Added "Energy Association" to windows menu.
 - ii. Export menu for ASCII tables now calls 'export_images_csv' with more popup options.
 - iii. GeoPIXE entry point now accepts argc, argv command-line arguments from OS and calls 'evt_start_do_commands'.
- n. Export_images_csv'
 - i. Removed /tab, select, /clip_zero options.
 - ii. Now uses two sets of popups to select between "XY" and "Table" mode exports.
 - iii. "XY" mode has new options to collapse onto a non-E axis, if a proxy_axis is found.
 - iv. ProjectX and ProjectY will always collapse if collapse is NOT set.
 - v. Uses an energy axis in export gable (from 'px_coords or 'py_coords') if found.
- o. Evt_start
 - i. Pass 'translate_file' and 'proxy_axis' to 'da_evt' for both cluster and non modes.
 - ii. Optional keyword /gencom to generate a GeoPIXE Command File (.gcf).
 - 1. Only works for
 - a. Sort_mode=0 (2D mapping)
 - i. Xy_mode=0 (normal XY scanning)
 - 1. mode=0 (**da_evt**), mode=1 (**cut_evt**)
 - 2. Not STIM mean energy
 - ii. Not X or Y step modes
 - b. Not sort_mode=1 (EXAFS spectra)
 - c. Sort_mode=2 (3D mapping)
 - i. Xy_mode=4 (XYE XANES 3D)

1. mode=0 (**da_evt_stack**)
 - ii. Xy_mode=5 (XYθ tomo 3D)
 1. mode=0 (**da_evt_tomo**)
 - iii. Not X or Y step traverse modes
- iii. Remove 'obj' argument to 'evt_start_image_increment2'.
- iv. Now use 'cluster_merge_images' to manage image stripe merges.
- v. Removed redundant setting of *pp parameters, such as pileup, throttle, linear files, all done in 'da_evt'.
- p. Da_evt
 - i. Optional "files=" keyword added for uniformity across GeoPIXE command file execution.
- q. cut_evt
 - i. Optional "files=" keyword added for uniformity across GeoPIXE command file execution.
 - ii. Added "cuts=" keyword as optional input for 'pcuts'.
- r. Da_evt_stack
 - i. Optional "files=" keyword added for uniformity across GeoPIXE command file execution.
 - ii. Optional 'select=' keyword as optional input for 'el_select'.
- s. Da_evt_tomo
 - i. Optional "files=" keyword added for uniformity across GeoPIXE command file execution.
- t. Geopixe_execute (NEW)
 - i. Calls 'execute', but to handle very long command-line strings, chops up into parts for params and keywords and 'executes' these first.
 - ii. Note this does not work under IDL VM.
 - iii. /progress keyword to enable progress pop-up for immediate (non cluster) mode.
 - iv. Add keywords for "spectra" and "images" to pass back pointers to spectra and images, if these keywords appear in the executed command.
 - v. NOTE: Some pointer args will not be freed, if not passed back. This may be bad for local execute. But this only happens for (transitory) command-file execution.
- u. cluster_merge_images (NEW)
 - i. Do image stripe merges here now. Needs 'presults' and 'images' pointer args, rest are optional and are found from 'pp' data.
- v. cluster_merge_spec (NEW)
 - i. Merge region spectra across cluster stripes (for one pass for large region count).
- w. geopixe_gen_commands (NEW)
 - i. Takes an argument list (vector) and writes a GeoPIXE Command File (GCF) file.
 - ii. Uses "#" comment lines.
- x. geopixe_do_commands (NEW)
 - i. Called from 'geopixe' to collect/pass command line args.
 - ii. Takes the command line 'argc' and 'argv' inputs to IDL and interprets these as:
 1. argv[0] command file
 2. argv[1] optional file list to replace "files=" line in GCF file.
preceded with "@" to supply a file-name containing the file-list, with one input file-name per line.
 3. argv[2] optional 'output' file-name to replace "output=" line in GCF file.
usually, both argv[1] and argv[2] will be supplied together.
 4. File list should be in stringify format. Will there be a character limit? If so, then may need the alternate approach (e.g. "@file", where "@" signifies reading the file list from this "file").
 5. cluster=0,1 to override setting in command file.
 - iii. Assumes that called commands use "files=" and "output=" keywords.
 - iv. For images, simply uses 'cluster_client' or 'geopixe_execute' and 'cluster_merge_images'.

- v. For spectra, needs to loop through 'cluster_client' or 'geopix_execute' and 'cluster_merge_spectra' to process groups of regions, then write all at end.
- y. Cluster_client
 - i. Only sets /floating mode, if 'group' is valid ID.
- z. Image_table
 - i. Add "C*" button to Extract tab generate a GeoPIXE Command File.
- aa. Geopix
 - i. Entry point now collects argc, argv and runs 'geopix_do_commands'.
- bb. Version 7.6k
- 2. Parallel_worker
 - a. Geopix_parallel
 - i. Calls out to 'geopix_execute' to process very long command string and do 'execute'.
- 3. Maia device
 - a. Allow YLUTs with only one blog file-name supplied.
- 4. Fortran
 - a. Da_accumulate11
 - i. Like 'da_accumulate5', but with multiple energy planes in 'matrix'.
 - ii. Pass in 'z' vector of energy plane index and 'n_energy' of third dimension..
 - b. Lib version 44

21 Apr, 2021

- 1. GeoPIXE
 - a. Spectrum_routines:
 - i. Current_plot
 - 1. Save current plot in common: c_spectrum_current, ispec_current for plugins to be able to access (to distinguish highlight and colour plot mode E cal).
 - b. Peak_Mn_centre_spectrum plugin, Peak_Cal_cut_centroid spectrum plugin:
 - i. Use 'current_plot()' stored in common (c_spectrum_current, ispec_current), to get correct E cal for current plot.
 - c. Spec_evt
 - i. Clip range of mask0 contributions to mask array within bounds for both
- 2. Scan_list
 - a. Set initial timer enable/time AFTER xmanager call.
 - b. New test for IDL 8.8+ and py3.6+ or IDL 8.5.1 and py2.7.

16 Mar, 2021

- 1. GeoPIXE
 - a. Fit_setup
 - i. Use 'weight_da_mode' in pars struct to trial various MPDA options in 'calc_da_matrix2'.
 - ii. See Confluence (run 9000444) and doc: "MPDA debugging ...".
 - b. Calc_da_matrix2
 - i. Refine 'weight' spectrum to avoid zero channel count and match 'f' spectrum max.
 - c. Define
 - i. Add "Learn" to /plot_options.
 - d. ImageRGB_eventcb
 - i. ImageRGB_export
 - 1. If plot.learn.on read in a Learn .rgb.csv file.
 - 2. Construct filename using (*p).file, not (*pstate).file
 - e. Plot_rgb_images
 - i. Loop if options.learn.on and (*pstate).plearn array present, as index iLearn.
 - ii. No hourglass as yet.
 - iii. Use (*(*pstate).plearn)[iLearn].r for Red, similar for Green, Blue
 - f. Plot_rgb_image_select
 - i. Add Learn cgcheckbox and file selection box.

- ii. Correct some errors found in plot_RGB_images_select_update and consistency elsewhere for options vector.
- g. Load_plot_options
 - i. Added learn and now have version -5 treated.

16 Feb, 2021

1. GeoPIXE
 - a. Batch_sort
 - i. In Notify 'done-filter', enable TIFF save for any of TIFF options 8,9,10.
 - b. Calc_da_loop
 - i. Only calculate 'rIntensity' if 'array.on' is true, i.e. for an array detector.
 - c. Load_geopixe_image_stack
 - i. Detect and report image load errors better.
 - ii. Needs manually selecting "Numeric" or "Numeric Part" at times for correct energy ordering in XANES.
 - iii. If DAI file name is tagged (relative to dir name), or if there are multiple forms (in first directory), then pop-up requester to select form to use for all energies/planes.
 - d. Strip_file_m
 - i. Allow multiple file processing (but not for /element option)
 - e. Scan_dir_evt
 - i. For multifile devices with multi_char() not '.', need to remove DAI filename ending:
 1. Ending = obj->multi_char() + ((adc_offset_device(obj) eq -1) ? '0' : '1')
 - ii. Corrected error to remove .dai. files from output list.
 - f. Evt_set_evt_file
 - i. For multifile devices with multi_char() not '.', need to remove segment number from deduced DAI filename ending:
 1. Ending = obj->multi_char() + ((adc_offset_device(obj) eq -1) ? '0' : '1')
 - g. EVT notify 'root':
 - i. Also remove segment number from deduced DAI filename ending.
 - h. Version 7.6h
2. Scan_List
 - a. **Changes made to source3, not done here yet.**

19 Jan, 2021

1. Scan_List
 - a. Source3 now uses the index marks and new REQ errno, which source2 does not YET.
 - b. Source3 also has the new errno for stage REQ.
 - c. **When this is working, and Perth has linear encoder setup, update scan_list.pro to source2.**

11 Nov, 2020

1. GeoPIXE
 - a. Image_table
 - i. Charge modify Erode/Dilate kernel sizes to "Erode 1", "Erode 2", ...
 - b. Image_table events
 - i. OnButton_Image_Table_Modify'
 1. When do 'Modify', then also - notify, 'image-region-update-one', (*pstate).pselect to force an update of all related pixel data: q, charge, IC_Count, conc, error, mdl, (sd relsd), centroids, phase, ayield.
 2. Charge Erode kernel sizes to give true 1,2,3... pixel erode all round.
 - a. Kernel of 2 only erodes on low side.
 - b. Kernel size = 2*erode+1
 - c. Write_geopixe_zarr (from Geopixe3)
 - i. Separate "channel" and "chan_active" axes.
 - ii. Make 'xc', 'yc' 1D arrays.
 - iii. Check for NaN in xc, yc?

- iv. Version 3
- d. Open_vsub: req_vsub
 - i. Include return 'errno' if python attribute 'code' is present.
- e. Fit_setup
 - i. Add 'weight_da_mode' to (*p).
- f. Calc_da_matrix2
 - i. Detect unity weight based on '(*p).weight_da_mode'
- g. Image_Process_Missing_Line
 - i. Now use 'tri_surf' for bilinear interpolation across a missing row/column gap.
 - 1. 'griddata' did not work with regular grid, as 'triangulate' fails.
 - 2. 'min_curve_surf' was way too slow and memory hogging.
 - 3. 'krig_2D' was not tried.
- h. Plot_rgb_images
 - i. Need to call 'set_device' with /True for RGB images in /PNG.
- i. Version 7.6e
- 2. Move_blog image plugin
 - a. Moves blog files from a second "append" run to continue the segment numbers of the first run. Needs "sudo" privilege and group RW access.
- 3. Scan_list
 - a. Start timer next:
 - i. Do a 'scan_list_maia_stop' regardless of error state.
 - b. If a warning popup is up, do other timer loops continue?
 - i. No it stops them!
 - ii. This will cause a Master to get hijacked.
 - iii. Make all plain 'warning' calls 'warning, timeout=300' calls.
 - iv. Make transfer of Master timeout 600s.
 - c. Scan_list_start:
 - i. Make phase0 up to save to KVS
 - ii. Report 'errno' from req_vsubs.
 - iii. Only change 'changed' if (*pstate).scan_sequence.lock_paused changes
 - d. Scan_list_next
 - i. Add a 'Phase0' to clear KVS active flag.
 - ii. Include req_vsub 'errno' return testing ...
 - iii. Only change 'changed' if (*pstate).scan_sequence.lock_paused changes
 - iv. Use 'errno' return to take action (prereturn, finish, error messages).
 - v. Add a separate phase for Maia start, so only one 'newrun'.
 - e. Scan_list_pause
 - i. Use 'errno' return to take action (prereturn, finish, error messages).
 - f. Version 7.6e

22 Oct Sep, 2020

- 1. Scan_list
 - a. Layout – new 'master' mode
 - i. Some spacing was wrong, missing mkatching entries in top of event loop.
 - b. Use a "master" hand-over approach to allow switching between Scan-List sessions and users and back again).
 - i. Only the master can control the stage and scans.
 - ii. The 'master_id' (node:user-seed) is stored in the state in KVS.
 - iii. Can 'hand back' control after a brief intervention (like NMP DAQ control case) to a 'pushed_master' saved there.
 - c. Now use 'scan_sequence' from 'state' **Hash** in KVS, which is retrieved first, and written back if changed (e.g. state 'MM.Per.SL.2.state.hash').
 - i. 'master_id'
 - ii. 'pushed_master'
 - iii. 'frame_id'

- iv. 'scan_sequence'
- d. scan_sequence:


```

active:      0, $           ; scan list is running
time_start: systime(1), $ ; time scan started
pause:      0, $           ; raster paused state
lock_paused: 0, $         ; keep the stage paused
move: { seqno: 0L}, $      ; 'seqno' sent for move
scan: { index: 0, $       ; present scan list index
      seqno: 0L}, $       ; 'seqno' sent for scan
raster_on:  0, $          ; raster started and 'running'
progress:   0.0, $        ; % progress through a scan
error:      0, $          ; error in current scan
stroke:     0L}, $        ; last completed stroke (use for "Append")

```
- e. Use KVS for scan sequence
 - i. Stage frozen alarm is tested for, even if active not 1.
 - ii. Add 'config_key', 'local_id', 'master', 'master_check' to pstate.
 - iii. Add "master-check" checkbox and event code.
 - iv. Add 'changed' to pstate to flag a change to 'scan_sequence' that needs to be 'put' before accepting any more 'gets'.
- f. If scan is running ('scan_sequence.active'=1), then KVS list takes precedence over local
 - i. If become master, and scan running, then /force read of /frame, i.e. scan list.
 - ii. This should make sure that becoming 'master' gets all scan rows 'active' status correct.
- g. New "Beam" LED
 - i. Shows Green = beam on and shutter open, Red = beam low or shutter closed, Off = Excillum off.
- h. Event loop
 - i. Notify, commands, timer: all use 'Scan_list_test_master', or 'Scan_list_get_state', which tests KVS, master and loads 'scan_sequence'.
 - ii. The old 'check_KVS' is now done within these routines, which is neater.
 - iii. Take care not to bypass the **timer renewals**.
 - iv. Only do the read state in the "Start" loop, to avoid overloading. In the other loops, just do the check_kvs (now as 'scan_list_check_kvs').
 - v. Add log for shutter open/close (Pause) and for Maia pause (event.enable).
 - vi. Set local flag 'changed' when any local scan_sequence parameters changed. Only allow further 'gets' (or test masters) after a successful 'put'.
 - 1. The 'put' clear the flag regardless of any errors to avoid a lock-up.
- i. Scan_list_test_master:
 - i. Test that KVS is OK, else return err.
 - ii. Test KVS master and set 'master' flag
- j. Scan_list_become_master
 - i. Become 'master' for scan sequence control
 - ii. Update scan list to set Green
- k. Scan_list_relinquish_master
 - i. Relinquish being 'master' for scan sequence control
 - ii. Update scan list to clear Green
- l. Scan_list_get_state, /frame
 - i. Retrieve the state hash from KVS for scanning
 - ii. If /frame check that current frame matches KVS.
 - iii. Check UTC time for scan-sequence, and if stale 'assume' Master.
- m. Scan_list_put_state, /frame
 - i. Write the state hash to KVS for scanning, if master
 - ii. If /frame put current frame ID in KVS.
 - iii. Make sure 'utc' is PUT at least every 10s even if nothing changed.
- n. scan_list_maia_scan
 - i. Filter out any quotes in comment, region text before write to metadata
- o. Scan_list_start, scan_list_stop, scan_list_next
 - i. Clear 'scan_sequence.lock_paused' to prevent lockups.

- p. Scan_list_shutter_pause
 - i. Log shutter close/open
- q. *pm – ‘power’ LED becomes ‘ready’
 - i. Add to status: bias, temp, bpinterlock
- r. Layout
 - i. “Power” LED becomes “Ready” with new context help.
- s. Event loop
 - i. Coords loop
 - 1. Read Maia bias, temp, bpinterlock in ‘coords’ loop and set (*pm).status.
 - 2. Take care as power and Maia status can be -1 (offline).
 - 3. Set power LED (now called “Ready”) for Maia, power and shutter
 - ii. Start loop
 - 1. Test Ready = power, shutter open and Maia for a Pause
 - iii. Add write of stage status return to Help text box (Scan, Move).
- t. Project
 - i. Load, save – import/export.
- u. Lock state
 - i. Make lock off the default
- v. Frame ID
 - i. Read the current ‘frame_id’ from KVS state on startup.
 - 1. But does not actually load table, because lock=1.
- w. Frame button
 - i. Now called “Scan Frame”, will always load current ‘frame_id’, not number of the text widget.
- x. A switch to lock=0 or master=1 will load frame_id
 - i. If master=1 and scan active=1.
 - ii. Note: does not work for going to lock=0 when master=0, because active=0 is forced.
- y. Pause action
 - i. Don’t use persistent REQ, as it loops anyway.
 - ii. Add logging for reason for pause.
 - iii. Put does not Test Master.
 - iv. **Pause ignores ‘no-op’ errors.**
- z. Scan_list next
 - i. Remove ‘req_vsub_persist’, as these would loop and undo the ‘utc’ heartbeat, etc.
 - ii. Now does ‘preturn’ to loop back via ‘skip timer’.
 - iii. Use warning with timer (and cancel to retry KVS and abort).
 - iv. Only set ‘changed’ if REQ succeeds (move and raster).
 - v. **Still needs to ignore no-op errors (e.g. already raster, pause).**
 - 1. See how it’s done in ‘scan_list_pause’.
 - 2. Need better error codes?
- aa. Scan_list_start
 - i. If move to origin fails 100 loop, then error and finish.
 - ii. Only set ‘changed’ if REQ succeeds (move and raster).
- bb. Version 7.6a
- cc. Scan_edit
 - i. Allow a blank ‘project’ name to be set
- 2. Maia-Control
 - a. maia_launch_interlock
 - i. If interlock status bad, then immediately read ‘bpinterlock.downtime’ to report correct downtime UTC.
- 3. GeoPIXE
 - a. Open_kvs and routines
 - i. Each ‘get_kvs’ does a ‘exists_kvs’, which doubles reads and time.
 - ii. May be better, in future, to skip the exists and let it force an error instead.
 - b. Str_remove

- i. Fixed a bug when removes used multiple times.
- c. Evt
 - i. Default for device droplist from 'geopixe.conf' file, via 'geopixe_defaults()' call.
- d. Write_geopixe_zarr (from Geopixe3)
 - i. Put all arrays at root level.
 - ii. Put all attributes in root level .attr, but use dot notation.
 - iii. Make 'elements', 'pactive' and 'mdl' arrays.
 - iv. Add attributes for "coordinates", "long_name" and "units"
 - v. Add 'xc', 'yc' arrays of stage X and Y coordinates, pointed to be "coordinates" attributes in relevant arrays.
 - vi. Fix missing "elements" in image and set Chunk=1 for element.
 - vii. Set element dtype to 'str'.

12 Oct Sep, 2020

1. Due to Eclipse incompatibility, and IDL-python version compatibility, must keep GeoPIXE2-source2 (IDL 8.5.1/8.7, python 2.7) separate from GeoPIXE-source3 (IDL 8.8, python 3.8).
 - a. GeoPIXE-source3:
 - i. Move it to version 8.6 (to parallel source 2 at 7.6).
 - b. GeoPIXE-source2:
 - i. Move it to version 7.6
 - c. GeoPIXE-source:
 - i. Maintain at version 7.5+
 - ii. Copied from GeoPIXE-source2 (take care with any builds – correct target dir).
2. GeoPIXE
 - a. Version 7.6.
3. Maia
 - a. Version 7.6

21 Sep, 2020

1. GeoPIXE
 - a. write_geopixe_zarr
 - i. Added the attribute "_ARRAY_DIMENSIONS" to all arrays for compatibility with Xarray. This lists the axis names (in reverse, or "C", order).
 - ii. Changed the "El" root attribute to "element", listing all element names.
 - b. File_requester
 - i. Made changes that enable /dir /write usage to write a new directory.
 - ii. This does not make the directory, but allow passing back a new path name.
 - c. Draw_spectrum
 - i. To 'show_diff', need to map spectrum from its 'cal1' to fit's 'cal' before subtraction.
 - ii. Scale all overlays by cal ratio 'cal1'/'ca', where 'cal1' is raw spectrum ca, to account for compression in fit.
 - d. Read_DA_as_spec
 - i. Adaptive scaling to make all matrix rows and pure spectra at least 10,000.
 - e. Read_rgb_images
 - i. Bug – missing 'opt1' for normal image (not 'alt').

21 Aug, 2020

1. GeoPIXE
 - a. Python_version
 - i. Returns python version string, and optional 'revision' major.minor (e.g. "2.7", "3.8")
 - b. write_geopixe_zarr
 - i. Write GeoPIXE image (pointer p) to a .zarr format dir tree.
 - ii. This needs python 3.4 at least.
 - c. Image
 - i. Test python_version, and if python not 2.7, disable KVS and MM open.

- d. Write_geopixe_image
 - i. Uncomment delete of .dai.* files.
- e. Image
 - i. Test python_version, and if python not 2.7, disable KVS and MM open.
- f. Extend RGB Image for 2x Compare images
 - i. imageRGB_eventcb:
 - 1. compare3, save GIF + JPEG, learn, select-element, select-max
 - 2. post-create base: add palt2
 - ii. imageRGB_routines
 - 1. draw RGBs, legend RGB, make TVB, RGB view
 - iii. plot_RGB_images
- g. interelement
 - i. Only notify 'image-display' (to Image, RGB image) if the image has been "changed".
 - ii. After filter, only use source > 0
- h. Corr routines
 - i. Make_tvb_corr
 - 1. Get ix, iy, image from "interelement_transform", if modified in interelement_operations.
- i. Da_evt
 - i. Bad test forming 'mat_inv_yield' (just testing less than 1.0e-5)
 - ii. Now just use 'finite()'.
 - iii. Divide by zero (NaN) also tested on output of 'yield' matrix, and Warning issued if these are found (e.g. occurs in non-element layers).
 - iv. These may be upsetting the "Compton" overlays?
- j. Read_da
 - i. 'read_da' for an mpdam file, loops through phases, and makes sure the element lists match, or selects the correct element index. For the case of "Compton", these occur in duplicates, one for In and one for Sn, in the MM case. Need to make sure that we select the first to match the first and the last to match the last. Fixed 'read_da' multi-phase loop to do this.
 - ii. This fixes both DA matrix rows and pure spectra, as well as yields, which effects both DA sort and spectrum reconstruction for MPDA case.
- k. Analyze_image
 - i. Same issue was fixed in the 'use_yield' = 1 case, to pick index 'k' that matches correct "Compton", or any duplicated name.
 - ii. This corrected both using the incorrect pure spectrum index, and which uses the yields.
- l. Append_da_fit
 - i. Disable the 'optimize_back' option to tweak the background strength for a better chi-squared fit.
- m. version 7.5+
- 2. Maia_launch
 - a. Test python_version, and if python not 2.7, disable KVS and MM open.
 - b. version 7.5+
- 3. Scan_list
 - a. Test python_version, and exit if python not 2.7.
 - b. version 7.5
- 4. Install
 - a. Output T&C to home .geopixe dir.

19 Aug, 2020

- 1. After install of IDL 8.8.0, update all GeoPIXE-source3 to GeoPIXE-source2
 - a. **NOTE: Eclipse environment for 8.8 may not be compatible with 8.7/8.5.1**
 - i. That's why this copy, and may not be able to run source3 with 8.5.1 (8.7.3?) after it has been "converted" for 8.8.

- ii. Remove Bat files for older versions from source 3.
 - iii. **Will need to maintain GeoPIXE3 and GeoPIXE2 source in parallel.**
 - b. Update docx and source code.
 - c. Release GeoPIXE2 using IDL 8.5.1, and to pftp.
 - d. Keep GeoPIXE2 as Version 7.5+
 - e. **Take care as destination for builds may still point to GeoPIXE3 ...**
2. Notes:
- a. Python 3
 - i. IDL 8.5.1 supports up to python 3.4?, while IDL 8.8 supports python 3.8.
 - b. Eclipse
 - i. IDL 8.8 changes some things and environment gets “converted”.
 - ii. May not be able to run GeoPIXE-source3 with IDL 8.5.1 after “conversion”?
 - c. Linux
 - i. Mr-05-mel
 - 1. Currently has python 2.7.17 and 3.6.9 (with zarr).
 - 2. IDL 8.5.1 will not run with python 3.6.
 - 3. Murray has installed python 3.4 as well.
 - Windows
 - ii. Anaconda3
 - 1. Python version 3.8.3 (zarr installed)
 - 2. Conda version 4.8.3
 - iii. Anaconda2
 - 1. Python version 2.7.13 (does not have zarr yet)
 - 2. Conda version 4.3.21
 - d. To get ZARR working on PC, need python 3. Using Anaconda3 (3.8.3) demands using IDL 8.8. This can be used on the PC. But we can’t deploy 8.8 SAV files to Linux as yet.
 - i. There is a “compatibility mode” approach to install ZARR under python 2.7.18.
 - e. If we want to use ZARR on Linux using IDL 8.5.1, then we need an older python 3.4 or use python 2.7.18.

13 July, 2020

- 5. GeoPIXE
 - a. Interelement_Operations
 - i. Skip out if *p does not have ‘modify’ struct (i.e. stack image).
 - b. Utc_from_date
 - i. Inverse of ‘date_from_utc’, with fractional seconds.
 - c. Date_from_utc
 - i. Error in “Jul” month string fixed.
 - d. Image
 - i. Remove menus for Clip to Right or Top.
 - 1. Only have “Crop to Shape” now, as it does it all correctly, to fix count-rate maps, etc.
 - 2. Have History “Cropped” not “Clipped”
 - ii. Add nenu for “Crop to Shape (zero outside shape)”
 - 1. Use /zero in crop call to zero
 - 2. Add extra Histrory to show “Zeroed outside shape”.
 - e. Image_table_EVT:
 - i. Popup warning if no YLUT table found, else cluster processes fails with a warning, which is a bit cryptic.
 - ii. Don’t need to build master mask, as this is done in spec_evt now.
 - f. Spectrum_display
 - i. Add menus for export of JPEG, PNG.
 - g. Plot_spectrum_select
 - i. Add options for /PNG, /JPEG
 - h. Plot_spectra

- i. Add options for PNG, JPEG using 'Z' device, passing landscape and portrait
 - ii. Remember to Erase before first plot.
 - iii. After plot(s), tvrd Z buffer /true, and write PNG or JPEG file
- i. Spectrum_select_eventcb
 - i. OnKill_select
 - 1. Only free pselect components, if it is a "good" pointer, with content.
- j. Corr_routines
 - i. Make_corr_tvb
 - 1. Only norm to charge map if element is NOT 'special'.
- k. Special_elements
 - i. Include 'Rates'
- l. Export_images_csv
 - i. Add /clip_zero switch (but default to 1)
 - ii. Only normalize to charge_per_pixel if ctype eq 0 (i.e. not a special plane, such as count-rate, dwell, etc.)
 - iii. Print as single formatted string line.
 - iv. Add a title line for XY tables.
 - v. Add menus to clip (or not) XY tables to zero.
- m. Image_correct_zero
 - i. Algorithm for correcting for zero pixels in images made a separate pro.
 - ii. Have indices vector 'q' as an argument
 - iii. Desired minimum number of good 'neighbours' to recover a 'zero' pixel as an argument.
- n. Image_eventcb
 - i. Image_Process_Zeroes
 - 1. Now calls 'Image_correct_zero', up to 3 passes.
 - 2. Threshold fixed at 3% above display Low (i.e. black pixels as displayed).
- o. evt_start_image_increment2
 - i. When accumulating count_rate_map, accept the larger of the terms from stripes (do not 'accumulate' rates).
- p. ratio_detector_yields
 - i. Remove ratios for tiny yields and divide by zero
- q. Map_spec
 - i. Add /quiet to suppress pop-up warning (returns err code only).
- r. Warning
 - i. Add cancel option to /info calls.
- s. Stay with version 7.5+ for now.
- 6. Raw_count_rate_image_plugin
 - a. Use "rate" not "c/s" for new 'element' name.
 - b. Using "c/s" (with "/") causes problems for labels of spectra (in "spectrum_index" routine).
- 7. Flatten_to_flux0 image plugin
 - a. Add options to select 'base' and what to do with "zeroes", such as to replace by interpolated values, using 'image_correct_zero'.
 - b. Use up to 3 passes to fix zeroes.
- 8. Depth wizard
 - a. Always prompt for depth table filename.
 - b. Add a "Clear" button to the results table tab.
 - c. wizard_depth_callback_calc_depths
 - i. Stop using /spline option in 'interpol', which seems to work better.
- 9. Install
 - a. If version 8.6 or beyond
 - i. Don't use Site Notice.
 - ii. Installation Number from 'lmgr2' now defaults back to "machine_name" from get_login_info().
 - b. Veto fixed Cisco VPN adaptor mac addresses

- i. Veto MAC addresses like: “00059a3c7...” (e.g. “00059a3c7800”, “00059a3c7a00”).
- c. Rebuild:
 - i. **geopixie_update, maia_update** (ftp_update build)
 - ii. **IDL_Query**
- 10. Running runtime GeoPIXE
 - a. **Python bridge issues prevent running GeoPIXE.sav, compiled under 8.5.1, under IDL 8.7.3.**
 - i. The “Runtime GeoPIXE 3.bat” file works using IDL 8.5.1 as this is how the SAV file was compiled.
 - ii. Run other runtime SAV files using the “2” scripts” or “3”, but using IDL 8.5.1.
 - b. Later experiment with IDL 8.7.3 compile and execute.
- 11. Maia_launch
 - a. Start a new log file each (UTC) day:
 - i. Pstate
 - 1. Pchart: add “time0:0.0d+0”
 - ii. Maia_launch_update_chart
 - 1. Create new file when date UTC changes.
 - a. Init (*pc).time0 when new file opened.
 - 2. Use delta time (relative to (*pc).time0)
 - 3. Add UTC date column to title and printout.
 - 4. If append to log and (*pc).time0 is not set
 - a. Read log file and get UTC from date-time on first row to set (*pc).time0.
 - iii. Maia_update_parameters
 - 1. *pc: add “time0:0.0d+0”
- 12. Maia_client_parameters
 - a. Update (*pm).version.software from ‘config.ps.version’

3 July, 2020

- 1. GeoPIXE source versions ...
 - a. Copy GeoPIXE-source3 → GeoPIXE-source2 (preserved 7.5+)
 - b. Copy GeoPIXE-source2 → GeoPIXE-source
 - c. Edit bat, prefs files for consistency
- 2. GeoPIXE release runtime versions ...
 - a. Copy GeoPIXE3 → GeoPIXE2 (preserved 7.5+)
 - b. Copy GeoPIXE2 → GeoPIXE
- 3. Save source and runtime (7.5+) in archives.

16 June, 2020

- 1. GeoPIXE
 - a. Spectrum_Mark_Pileup_Image
 - i. **Hardwire some “pileup deficit”** into the modelling of a pileup spectrum (-0.2%).
 - ii. Use ‘map_spec’ to map pileup spectrum (pure cross convol) on spectrum E cal.
 - b. Pixe_fit
 - i. Use ‘map_spec’ to map pileup spectrum according to ‘sum_deficit’, correcting for the -0.2% assumed above.
 - c. Spectrum_select
 - i. OnButton_Delete_Zero_Select
 - 1. Add delete mode to delete ‘zero spectra’
 - 2. This routine detects spectra with less than 1 count to delete.
 - d. Define:
 - i. /Image
 - 1. Add Modify struct, for use with interelement adjustments.
 - 2. Only one element modified at a time.
 - 3. Is NOT saved or read from disk.
 - 4. Must be “Apply”ed to be saved.

- e. `Interelement_Operations`
 - i. Pop-up to permit inter-element operations, e.g. subtract an artefact of one element leaked across into another.
 - 1. Uses a new 'modify' struct in the image struct to store temporary modifications.
 - 2. Only one element modified at a time.
 - ii. Must be "Apply"ed before being saved, or another element investigated.
- f. `Image_routines`
 - i. `Make_tvb`
 - 1. Call 'interelement_transform' to get 'img' if modified, else ignore.
 - 2. Only 'image' window shows dynamic changes before Apply.
 - 3. Musy "Apply" to see changes in RGB window.
- g. `OnNotify_Image`
 - i. 'image-process'
 - 1. All these now will contains 'mode' and 'name'
 - a. Added to 'OnSelect_Image_Process', 'OnButton_Image_Process_Get', 'OnButton_Image_Process_Undo'
 - 2. Add 'inter-element', which uses 'interelement_filter' and 'interelement_transform' from 'interelement_operations'.
 - ii. `OnButton_Image_Process_Get`
 - 1. Add 'inter-element' detection in image file and pass on 'image-process' with extra terms in struct: {mode:1, image:i, name: 'inter-element', valid:1, operation:'op name', arg1:scale*strength, filter:'filter name', arg2:'src element' }
- h. `Image`
 - i. Compile 'interelement_operations'
 - ii. Add "Crop to Shape" menu
- i. `Time_amplitude`
 - i. Make station-text change set slider.
 - ii. Add arrow (Linux only) to increment detector +1.
- j. `Image_table_evt`
 - i. Use 40% of 'default.memory' from `geopixe_defaults()` as maximum for mask memory.
- k. `Geopixe_defaults()`
 - i. 'default memory' added (bytes) to 'geopixe.conf' file.
 - ii. Update template 'geopixe.conf' file.
 - iii. Defaults to 20 GB.
- l. `Image_Process_Crop`
 - i. Crop to any shape extremities
 - 1. /zero: Also zero image (and flux) outside of shape, within these bounds.
 - 2. Warn if `xcompress,ycompress` not 1 (can't do simple crop then)
 - ii. `sub_region=1`, Set `xoffset,yoffset`
 - 1. Set `xsize,ysize, xsub_range,y_sub_range` to new size
 - iii. Clip data:
 - 1. Bounds,
 - 2. Clip maps: `flux, raw_flux, dead_fraction, dwell_map, pileup_map, count_rate_map, error, image`,
 - 3. Clip: phase, yield map (yield, size of error), note different number of planes.
 - iv. Charge, scale by pixels to fraction of total charge
 - v. Temp: set `valid=0`
 - vi. Preview: free (*p).preview, has_preview=0) to force new one made on write
 - vii. Not corrected:
 - 1. Invalid: processed, clipped, valid
 - 2. What about: `px_coords, py_coords`? Leave for now?
- m. `Veto`

- i. If (max) source 'q' is already -1, then return, -1.
 - n. Evt
 - i. Do not set (*pstate).path for Get Cal or read DA matrix file as these are often stored elsewhere in 'config'.
 - o. Image_history
 - i. Pass on 'show' element index to pstate.
 - p. Image_table
 - i. Extra 'Save' on hotspots tab
 - q. Image_save_GIF
 - i. Added /JPEG option for JPEG output (quality=100).
 - r. Version 7.5+
2. Standards wizard
 - a. If open run # range used, it set maximum to just 1,000,000, which is not high enough for Perth runs (e.g. #9000123). Fixed.
 - b. Version 7.5.
 3. Maia_control
 - a. build_kandinski_script_step_cal
 - i. After round to make steps whole for a 10 bit DAC, need to adjust count to compensate, otherwise High becomes too small.
 - b. Use default detectors from config file – 96 allows no DDM.
 - i. Dodge errors and set flag for no DDM.
 - c. Version 7.5+

12 June, 2020

1. GeoPIXE
 - a. Image_table
 - i. Image_table_load
 1. When load matching spectrum, check number and correct labels.
 2. Notify 'image-regions' w/ pointer array, if regions AND spectra are valid.
 - ii. Image_table_evt
 1. Search updir=5 for blog data (needed for old Perth dirs: "blog/left/2018/run/")
 - iii. OnButton_Image_table_hotspots, neighbourhoods
 1. Reset index values after creating new regions.
 2. Clear preregions and send to Spectrum-Display (spectra don't match regions).
 - iv. OnButton_Image_Table_Clear, ...delete_all
 1. Clear preregions and send to Spectrum-Display (spectra don't match regions).
 - b. Evt_start
 - i. Search updir=5 for blog data (needed for old Perth dirs: "blog/left/2018/run/")
 - c. File_requester
 - i. If /skip_if_exists, then default to /read.
 - ii. If updir>0 or /translate (and write=0), then default to /read
 - iii. Make sure you don't use /read, if you do not want any search.
 - iv. Only do safe_make_dir if /write mode access to new dir.
 - d. Read_mpdam, read_correct, image_process_eventcb, image_eventcb, spectrum_display_eventcb, write_spec, standard wizard, depth wizard, spectrum_load_prep, retry_image_increment
 - i. Fixed options for 'file_requester':
 1. /read if search needed. No /read if no search desired.
 - e. Spectrum_load_do
 - i. Use explicit /read in calls to file_requester
 - f. Imgr2
 - i. Extract mac address from last entry on the line.
 - ii. Licensing now works under Windows using IDL 8.7.3.
 - g. Image
 - i. Image_results_table

1. Register 'image-regions' when launching image_table.
 - ii. Image_clone, spectrum_display
 1. Register 'image-regions' when launching image clone.
- h. spectrum_display
 - i. Register 'image-regions' when launching spectrum_display.
- i. Spectrum_display_eventcb
 - i. Spectrum_delete_select
 1. Fix bug to protect region #0, when /keep_zero.
 - ii. Spectrum_Mark_Pileup_Image
 1. Now use image products, just for the pixels selected in a region.
 2. Test that region is valid (matching spectra)
 3. Find region 'index' to match spectrum label "Region ...".
- j. Fix_path, dir_up
 - i. Allow string vector argument, return string vector.
- k. File_search2
 - i. 'Exclude' argument to maintain a list of all dirs tested, so they don't get done twice.
- l. File_requester
 - i. 'path' now can be a list of paths (use for [dpath,path] for data)
 - ii. Uses 'file_search2' passing 'exclude' to avoid testing dirs more than once.
- m. Version 7.5_
2. Maia_control
 - a. Read_maia_parameters
 - i. Tweak options to file_requester, as above.
 - b. Linearise2
 - i. Use Kandinski config.ps.version >= 9542 (instead of 9461).
 - c. Maia_setup
 - i. Enable: Toggle linearise2.enable now (set linearise.enable=0)
 - d. Version 7.5_

3 June, 2020

1. GeoPIXE
 - a. Image_table
 - i. OnButton_Image_Table_Modify
 1. If Erode results in no pixels, fallback to using a single pixel at the centroid position.
 - ii. OnButton_Image_Table_Export
 1. If multiple matches to hotspot element, use the lowest shell match.
 - iii. Image_table_EVT
 1. Evt_files undefined for non-Maia data, fixed (=evt_files).
 - b. Gprint
 - i. Make default 'default_gprint_unit' = -1 (stdout).
 - c. Da_evt, cut_evt, da_evt_stack, da_evt_tomo, ...
 - i. Make default 'cluster_debug' = -1 (stdout).
 - ii. Had already fixed 'spec_evt'.
 - d. File_search2
 - i. Make sure 'p' is not referenced if 'file_search2' is called without /progress.
 - ii. Use goto, finish before forming progress.
 - e. Evt_start
 - i. All file_requester reads should specify /read, and make all updir=4
 - f. Define
 - i. Add "project" to /maia_scan_spec
 - g. Geopixe_defaults
 - i. Add "projects" to "paths"
 - h. Version 7.5*
2. String_packer

- a. Hide_embedded
 - i. Explicitly treat levels of single and double quoted strings separately.
- 3. Scan_list
 - a. Scan_list
 - i. Update table shows project where region was, region moved towards end of row.
 - ii. Tweak options to file_requester, as above.
 - b. Scan_edit
 - i. Hide sample Ref1, Ref2 and sample coords mode.
 - ii. Add Project selection pop-up (no edit text).
 - 1. “?” will first load KVS “MM.project.list” and then read the ‘projects’ dir.
 - 2. Any new projects in dir will be appended to the KVS.
 - iii. NOTE: This only updates the local KVS, not the other one. But that will happen when Scan_List is run in the other place.
 - c. Version 7.5

2 June, 2020

- 1. GeoPIXE
 - a. Image_table
 - i. New organization of the buttons into tabs (Load, Modify, Delete, Extract, Hotspots), and add new feature (modify highlight pixel selections, with Erode, Dilate operators).
 - ii. Event routine: Delete button
 - 1. Pass keep_zero as well as select list to Spectrum Display
 - iii. New_table_pselect
 - 1. Return optional ‘select’ of all ‘index’ values in top-bottom range.
 - 2. Pass into ‘(*pstate).pselect’ the unsorted position in table to pass to Spectrum.
 - iv. OnButton_Image_Table_Modify
 - 1. Do Erode or Dilate of selected highlight map.
 - 2. Save previous in Undo buffer
 - v. OnButton_Image_Table_Delete
 - 1. Use /keep_zero arg to protect region #0 (unsorted index)
 - 2. If ‘select’ arg, then popup region # list to select.
 - vi. OnButton_Image_Table_Hotspot_Update
 - 1. Update the charge, IC_total and flux for region #0 (unsorted index)
 - vii. OnButton_Image_Table_Star
 - 1. Not used any more.
 - b. Spectrum_display
 - i. Notify: 'image-region-delete':
 - 1. Pass keep_zero as well as select list to Delete Select.
 - ii. Spectrum_Delete_Select
 - 1. If /keep_zero, then also accumulate charge, IC_Total for region #0 as sum of others.
 - c. Version 7.5^

16 May, 2020

- 1. Maia Control
 - a. Maia_setup
 - i. Simply read in .linear.var file and write ot out again to socket.
 - ii. If not .var, use the get_linear() function.
 - b. Maia_launch_read_enable
 - i. If either linearise.enable or linearise2.enable set, the set ‘(*pm).linear.on’
 - c. Version 7.5%
- 2. Maia_client_parameters
 - a. If either linearise.enable or linearise2.enable set, the set shared mem ‘enable.linear’.
- 3. Linearise2 spectrum plugin
 - a. Now detects which spectra have data in each chip.

- b. Automatically trims the ends to avoid saturation at top (as low as channel 4020).
 - c. Rejects end peaks, if they show strange totals (partial peaks).
 - d. Produces new linearise2 .linear.var output file.
- 4. GeoPIXE
 - a. Get_linearize
 - i. Can read new linearise2 var files.
 - ii. Returns multi=1 for old (ver=2) and new linearise2 (ver=4) files.
 - iii. Used:
 - 1. Export_da, export_da2
 - a. Does not know about multi det or fractional tables.
 - b. Expects F returned to be simple integral table for one det.
 - 2. Da_evt, spec_evt, cut_evt, xanes_da_evt, ...
 - a. Simple table or multi tables for many detectors returned.
 - 3. Maia_setup
 - a. Now only called for old .linear files.
 - b. New .var files copied into Kandinski direct.
 - iv. Details struct:
 - 1. Only used in Maia_setup, now only for old .linear fractional files (ver=3).
 - b. Batch sort
 - i. Scan_dir_evt
 - 1. Allowing a match of start of DAI file name to blog name relaxation would accept “-RT” realtime files as well, which is no good.
 - 2. Now excludes explicitly “-RT” after blog name in dai name.
 - c. Spec_evt
 - i. ‘start_mask’, ‘stop_mask’, index selection to process sub-set of regions list
 - ii. Fixed use of ‘i’ versus ‘qi’ in final wrap-up (see “;@5-20” comments):
 - 1. By_detector mode:
 - a. i = index into qmask, found detectors list
 - b. qi = detector number
 - 2. Normal region mask mode
 - a. i = index into qmask, accepted regions index
 - b. qi = i
 - c. qmask[qi] = index of accepted mask
 - d. pmask[qmask[qi]] = pointer to region mask
 - d. Image_routines
 - i. Analyze_image, read_table
 - 1. Set (*presults).index to position in table
 - e. read_table
 - i. Set results.index to position in table.
 - f. Define
 - i. /table
 - 1. Add ‘index’ for original position in table
 - g. image_table_eventcb: Process huge region masks and sort table by column values
 - 1. Sorting table creates issues for row delete, accumulated qc for region #0 (which moves elsewhere) and Notify to spectrum display for selection (where rows are not re-ordered).
 - ii. image_table_EVT
 - 1. Restore index order first before sort.
 - 2. Calculate maximum # regions for a 20 GB byte mask array across all processes, and process just this many max.
 - a. Need some global large array size limit in geopixe.conf?
 - 3. Loop until all regions are done, appending to spectrum pointer array.
 - a. Still slow, as need to test each event against ~500 masks.
 - 4. Ignore this for Xstep data.
 - 5. If ‘cluster_client’ returns error (e.g. Cancel), it breaks out of loop.

6. Write combined pp file and end of last pass.
7. Construct a master mask layer and save to a file, to pass 'master_file' to spec_evt.
- iii. Load_image_table_table
 1. Save 'col_index' (index back into conc) in pstate
 2. Add index to 'Image' column text with element(s).
- iv. OnCellSelect_Image_Table
 1. If whole column is selected, sort the p[] array on element conc, unless column zero, then sort using original index.
- v. New_table_pselect
 1. Pass sel.top as index ((*p[i]).index) to spectrum.
- vi. OnButton_Image_Table_Save:
 1. Restore index order before write
- vii. image_table_restore_index_order
 1. Restores index order
- viii. OnButton_Image_Table_Hotspot_Update
 1. Must now search for region #0 in list (if re-ordered)
- ix. OnButton_Image_Table_Delete
 1. Populate list with index number, not row #.
 2. Return 'select' as list of index values.
 3. Veto any with index=0, if /keep_zero
 4. If delete last row, set (*pstate).sel.top = i-1
- x. new_table_pselect
 1. Set sel.top = sel.bottom = index into list of 'index' values for current row, assuming these remain sorted in ascending order, as will be the case in Spectrum_Display.
 2. Preion is passed, pointing to p[] pointer for the selected row.
- h. Spectrum_display_eventcb
 - i. 'image-region-delete' Notify
 1. Calls 'spectrum_delete_select', /keep_zero with select list from Image-Table
 - ii. Spectrum_Delete_Select
 1. Uses 'sel' input (list or region# to delete), and searches these against region# obtained from spectrum labels.
 2. If /keep_zero, then construct spectrum for region #0 as sum of others.
- i. Var_type
 - i. Handle special case of Objects. 'typename' will return the Class name for a scalar object and "OBJREF" for an array or a null object, which is a pain.
 - ii. So, if name is not in the list of known objects (i.e. LIST, HASH, OBJREF, ...), then we'll assume that it's a Class name for an object of that type.
- j. Element_select
 - i. If close window, then return all zero /off
 - ii. /Off to set initial values as off.
- k. Spec_evt
 - i. New args:
 1. master_mask, master_file to use a mask master to speed processing
 2. start_mask, stop_mask to bracket regions
 3. Add "toc" at regular intervals (assumes "tic" done before)
- l. Spec_accumulate4
 - i. Add a new argument 'master' as a master mask array, which should be checked first before proceeding to check all masks for a match.
- m. Da_evt
 - i. Add "toc" at regular intervals (assumes "tic" done before)
- n. Version 7.5%
5. String_packer
 - a. Stringify

- i. Encode a object reference, simply, as `obj_new()`.
 - ii. Only do this if `/embed_ptr` is selected.
 - iii. For a valid Object, get the Class name using `obj_class()`.
- b. Unstringify
 - i. Decode a 'stringify' string with embedded objects and Obj arrays.
- 6. Geopixe_parallel
 - a. Start a "tic" to start timing. Add "toc"s for timing.
 - b. Added "toc"s in `spec_evt`, `da_evt`, using passed `cluster_debug=olun`.
- 7. Fortran lib
 - a. `Spec_accumulate4`
 - i. Add 2 arguments 'master', 'use_master' to check a master mask array first.
 - b. Lib Ver 43

13 May, 2020

- 1. Maia Control
 - a. `Maia_setup`
 - i. Apply imaging:
 - 1. For ROI, set `rGamma` size in array struct to `(*pm).DA.N` rows.
 - 2. For DA, explicitly make `rGamma` array of correct length, and fill it from 'da.array'.
 - b. `Maia_launch_clear`
 - i. Removed un-neccsary ET2D memory clear, as it was there already.
 - ii. Fix dimensions of this from explicit 2D (now wrong) to 3D (implicit using `[*]`)
 - c. `Blog_client_et2`, `blog_client_da2`
 - i. Reign in the logs about missing records. Now does not permit a second log until 100 s
 - ii. Avoid any print statements (e.g. in `gprint` or via `log_message`, warning), as these eventually fill up the stdout buffer and hang any further `log_message` output and stalls the process.
 - d. `Maia_launch`
 - i. Have setup string buffers to accept output from the background proceses, coming back from reads on the pipe LUNs.
 - ii. Setup framework for timers to trigger these reads – but commented out for now.
 - iii. Any reads from pipe (in `log_client`) hangs IDL.
 - iv. Now using IDL Bridge, so ignoring LUNs now.
 - v. Use `Obj_destroy` to end background processes.
 - e. `Spawn_blog_processes`
 - i. Tried variants of spawn command, but it does not work without the `/noshell`, `/hide` options (under Windows).
 - ii. Now using IDL Bridge call, via 'spawn_bridge_object' function.
 - f. `Spawn_maia_processes`
 - i. Now using IDL Bridge call, via 'spawn_bridge_object' function.
 - g. Version 7.5@
- 2. `Blog_client_da2`, `Blog_client_et2_spectra`, ...epics, ...groups, ...activity
 - a. Now use args: prefix, conf, server, debug
 - b. If debug, will write to a log file in .geopixe dir.
- 3. `Maia_client_parameters`, `Maia_client_parameters_slow`
 - a. Now use args: prefix, conf, server, debug, `n_detectors`
 - b. If debug, will write to a log file in .geopixe dir.
- 4. GeoPIXE
 - a. `Log_message`
 - i. The print statement eventually overflows the stdout buffer of background processes, which then stall during the 'log_message' call.
 - ii. Comment out the print statement.
 - b. `Log_client`
 - i. A routine to read from pipe from background processes.

- ii. However, it hangs IDL, so do a quick return for now.
 - c. Warning
 - i. Only do the gprint for output not zero (i.e. under debug conditions to a file).
 - ii. No, this does not work – frustrates the GeoPIXE build process.
 - d. Bridge_spawn_object
 - i. Creates an IDL Bridge object and ‘executes’ the background process in it.
 - ii. Does not have access to any print stdout I/O now.
 - e. Parallel_client
 - i. Init:
 - 1. Now uses ‘spawn_bridge_object’ to spawn processes.
 - 2. Saves object to ‘c_parallel_obj[]’ array.
 - ii. Monitor:
 - 1. Check status (obj->status()) for 3 (error halted) or 4 (aborted)
 - i. 1 – executing OK
 - ii. 2 – completed
 - iii. 3 – error halted
 - iv. 4 – aborted
 - b. But will these (3,4) get set, if error catching in worker pops up a warning on an error? Perhaps not?
 - 2. If these return monitor = -1
 - iii. Cancel:
 - 1. Use obj_destroy on c_parallel_obj
 - f. Version 7.5@
5. Parallel_worker
- a. Geopixe_Parallel
 - i. Pass args in as stringify(args).
 - ii. Args: workerIndex, totalWorkers, prefix
 - iii. Even if the execute completes OK (status=1), the error state may still contain the results of an EOF during processing.
 - 1. This would make the parallel_client – monitor see an error.
 - iv. Hence, if status=1 (success), use ‘message/reset’ to clear error flags before returning.

30 Apr, 2020

- 1. Maia Control
 - a. Maia_setup
 - i. Apply imaging:
 - 1. For ROI, set rGamma size in array struct to (*pm).DA.N rows.
 - 2. For DA, explicitly make rGamma array of correct length, and fill it from ‘da.array’.
 - b. Maia_launch_clear
 - i. Removed un-neccsary ET2D memory clear, as it was there already.
 - ii. Fix dimensions of this from explicit 2D (now wrong) to 3D (implicit using [*])
 - c. Blog_client_et2, blog_client_da2
 - i. Reign in the logs about missing records. Now does not permit a second log until 100 s have elapsed. This prevents it from swamping the logging system.
 - ii. Add a periodic log (every 100s) that summarises good, bad, lost records.
 - d. Blog_client_epics
 - i. Add the log_message, as in others.
 - e. Maia_client_parameters, maia_client_parameters_slow:
 - i. If log a Kandinski error, then wait 10 seconds, to prevent too many.
- 2. Maia_launch
 - a. Add heart beat LEDs to monitor the timer loops.
 - b. Chart error (‘maia_update_parameters3’, LED 2) would exit timer loop.
- 3. GeoPIXE
 - a. State_button

- i. Increase size of 'highlight' spot in centre by 1 pixel (to right, down).
 - b. Log_message
 - i. Also prints the message.
 - c. Spectrum_select
 - i. Also send Notify 'select-highlight' for table row selected, to go to Maia Rates
 - ii. Use existing pointer heap (pselect) in pstate
 - iii. Also do it for Prev, Next.
 - d. Spectrum_display
 - i. Needs to expect a 'select-highlight' Notify event from spectrum-select.
 - e. flux_flatten
 - i. change default for 'base' from 0.05 to 0.001
 - f. image_flux_flatten
 - i. Pass back 'scale' as return keyword result.
 - g. Image_correct_flux
 - i. Collect 'scale' from 'image_flux_flatten'
 - ii. Store in new 'scale' map in image struct
 - iii. Comment out for now (will require new version)
 - 1. Will need changes to 'define', 'read_geopixie_image' and 'write...'
4. Fortran
- a. Hist_accumulate2 (as called from IDL 'host_accumulate'):
 - i. Clip ET2D sum at: $\text{long}((2LL)^{31}-1) = 2147483647$
 - b. Lib Version 42

26 Apr, 2020

1. Maia Control
 - a. maia_setup_apply_imaging
 - i. Da.element[].scale is a fixed point (1.0e-6 - 1.e+23) and cannot handle smaller floats. While normal da.matrix rows have values $\sim 1.0e+7$, the Back1 and Back2 rows have value more like 1.0e-10 or less.
 - ii. Clipping the 'scale' factor at 0.001 restores display of Back1, Back2 maps.
 - b. maia_setup_apply_hymod
 - i. Failed to close 'lun' after read of new var gaintrim file, which used up inits after a few Apply Hymods.
 - ii. Now does a 'close_file, lun'.
 - c. Maia_update_da2
 - i. Set (*ppar)[10] = (*pm).scan.X
 - d. Maia_launch_clear
 - i. Also clear: temp total_flux, total_pixels, charge
 - e. Version 7.5z
2. Scan_list
 - a. Lots of errors in Log.
 - b. Some errors for reading "MM.Per.CF.newscan" and "MM.Per.GP.1.position" from Perth KVS.
 - i. Fixed this by copying the same data from Mel KVS.
 - ii. Also fill in matching variables "MM.Per.2.position" and in Mel too.
 - c. Scan_list_maia_scan
 - i. Pass 'flux'chan[0].name' as "Maia:dwell.time"
 - d. Version 7.2
3. Blog_client_...
 - a. Commented out the log_message and gprint of "timeout reading header", which tends to happen solely because we are waiting for a packet, not an error.
4. Blog_client_da2
 - a. Use ScanX from (*ppar)[10], if not zero, else keep using 'nx'
 - i. Scale down by compress.
 - ii. Use this instead of 'nx' for FlipX

5. GeoPIXE

- a. OnButton_image
 - i. Changed the adjustment of handle 4 (middle) to simply move and drag the side handles (9,10) too. It does not change the tapering. This is done solely using 9,10 with right mouse button.
- b. Time-amplitude
 - i. Added controls to tweak the T gaintrim parameters.
 - ii. Map back onto pre-gaintrim T using ptrim0 set (loaded from file) and then map onto tweaked parameters after adjustment to plot E-T within pileup field.
 - iii. Added menu to save new T gaintrim file and button to load initial T gaintrim file (used to acquire the blog data).
 - iv. Changed display to fixed 4096 x 1024 display, thick=1.0, csize=1.2.
- c. Get_gaintrim
 - i. New routine to load a gaintrim file (either .var or .spec).
- d. Et2d_image
 - i. Call 'simple_image' using /ganged
- e. Simple_image
 - i. New option /ganged, to force all detector/elements to be controlled at the same time for min/max and Z-scale mode.
- f. Version 7.5z

21 Apr, 2020

1. GeoPIXE

- a. Plot_images
 - i. If window 12 not open set an aspect ratio to match upcoming plot (as was done in 'plot_RGB_images'.
 - ii. For JPEG, PNG, call 'set_device' setting aspect=(float(sy)/float(sx)) so that it sets size dynamically every time.
- b. Parallel_client:
 - i. Add wait=1 after each spawn, to give IDL (8.6) on Linux time to deal with new license access.
 - ii. Increase overall timeout = 120.
- c. Plot_RGB_images
 - i. For JPEG, PNG, call 'set_device' setting aspect=(float(sy)/float(sx)) so that it sets size dynamically every time.
- d. Set_device
 - i. Juggled scale_z to scale as 1/sqrt(aspec)
- e. Image_table_eventcb
 - i. OnButton_Image_Table_Export
 1. In normal mode: using the conversion 'float2()' on forming 't2' caused it to become a linear array.
 2. Work-around using 'size()' and 'reform'.
 3. Removed this after fixing 'float2' (below).
- f. Hist_accumulate:
 - i. Still calls Fortran 'hist_accumulate2', but passes extra parameter: 'n_et2d'. Test dimensions of the 'et2d' array.
- g. Version 7.5y

2. Stringify

- a. Float2, long2, long2_64, fix2, double2:
 - i. Maintain array structure of input in output.

3. Parallel_worker:

- a. Geopixe_parallel:
 - i. Increase timeout = 100

4. Maia_control

- a. Maia_Launch – ET2D display upgrade to 3D

- i. N_energy = 256, N_time = 128
 - ii. Buffer_size3b = [n_energy, n_time, n_detectors+1]
 - iii. pimg3 = ptr_new(fltarr(n_energy, n_time, n_detectors+1))
- b. Maia_setup:
 - i. Add “Adj T gain” arrows and matching events ‘hymod-gaintrim-T-gain-up’, etc. to change gain by +/-2%.
- c. Version 7.5y
- 5. Blog_client_et2_spectra:
 - a. No changes or recompile needed as ‘hist_accumulate’ (loaded with GeoPIXE.sav) does all the work with the new 3D et2d array.
- 6. Fortran:
 - a. Hist_accumulate2
 - i. Pass ‘n_et2d’ as well, as extra argument.
 - ii. Increment individual detector maps, as well as sum map.
 - b. Lib version 41

11 Apr, 2020

- 1. GeoPIXE
 - a. Image_table
 - i. “*” Delete select option now uses ‘element_select’ to select regions to delete, but vetoes region #0 using /keep_zero.
 - 1. Does update of #0 qc automatically.
 - ii. OnButton_Image_Table_Star:
 - 1. Delete uses select and /keep_zero
 - 2. Does update of #0 qc
 - iii. OnButton_Image_Table_Delete
 - 1. Uses ‘select’ to select regions to delete, but vetoes region #0 using /keep_zero.
 - 2. Does NOT do ‘image-region-select’, which conflicts with notify later.
 - iv. Issue remains:
 - 1. Conc values in table still contain deleted regions.
 - 2. Also need to update conc table, some how?
 - 3. 'Flux' in conc table is still wrong.
 - b. Spectrum_select
 - i. ‘image-region-delete’ calls ‘OnButton_Delete_Spec_Select’ passing vector to delete.
 - ii. OnButton_Delete_Spec_Select
 - 1. Uses ‘select’ to select regions to delete, but vetoes region #0 using /keep_zero.
 - 2. If /keep_zero, then sums other spectra into #0
 - iii. Issue remains:
 - 1. Overlay is too big because conc values in table still contain deleted regions.
 - c. Version 7.5x

25 Mar, 2020

- 1. GeoPIXE
 - a. Back_update
 - i. Routine to read in all Back plugins and return filenames and titles.
 - ii. ‘present’ can be current filename or title and back_update will return new filename/sav name.
 - b. Fit_setup
 - i. Load back plugins using ‘back_update’.
 - ii. Fit_setup_do_fit:
 - 1. Fix the order of the priorities for plotting overlays (added comments).
 - c. Plot_rgb_images
 - i. Add ‘max_area’ option to minimize borders.
 - d. Plot_rgb_image_select
 - i. Add ‘max_area’ option to minimize borders

- e. Define
 - i. Add max_area to plot_options.
- f. Geopixe_library:
 - i. Switched off DEBUG ONLY code that loads 'image_dll.dll' for Windows.
 - ii. Should not have been on!
- g. Split_back
 - i. Check for a valid second pback, else assume single only.
 - ii. Check for valid pback pointer, *pback pointer array, *(*pback)[0] pointer and *(*pback)[0].data pointer to data.
- h. Image_routines
 - i. Make_TV
 - 1. For non screen output, if compress > 1, set 'nmin' = 2.
 - 2. This stops too much dilation of features if writing into large Z buffer pixmap area.
- i. Detector_show
 - i. Set target normal colour for b/w backgrounds.
- j. Layer_plot
 - i. Take care if a layered samples. Only choose microns scale if all layers have /microns set.
- k. Non_linearity3
 - i. Add 11th and 13th order polynomials.
 - ii. Make sure the function is performed completely in double precision.
- l. Batch_sort
 - i. Use /silent to avoid resolution pop-up in batch mode.
- m. Version 7.5w
- 2. Wizard standards
 - a. Resize now implemented using a table of X widget dimensions (and separately for Y). The table holds the ID and normal scr_xsize (scr_ysize). These are adjusted by the change in the TLB scr_xsize (scr_ysize).
 - b. *pstate includes initial TLB scr_xsize, cr_ysize and resize tables.
 - c. Version 2.5
- 3. Linearize spectrum plugin
 - a. Add 11th and 13th order polynomials.
 - b. Make sure the function is performed completely in double precision.

19 Mar, 2020

- 1. GeoPIXE
 - a. select_element_list:
 - i. More error catching.
 - b. File_requester
 - i. Was defaulting 'path' to path of file, which made it same as 'remote'.
 - ii. Now, if no 'path' supplied, then default local path to current dir (variable 'default').
 - c. Image_RGB_eventcb
 - i. Fix bug in batch that prompts for "full resolution" after every sort.
 - ii. OnNotify: 'batch-save'
 - 1. Call ImageRGB_Learn with /silent for /execute.
 - iii. ImageRGB_learn
 - 1. Skip pop-up and assume zoom=0, full=1 if /silent.
 - d. Version 7.5w
- 2. Install
 - a. Use lmgr2(), as in startupg, if IDL version 8.6 or later.
 - b. Version 8.7
- 3. Geopixe_update
 - a. Version 8.7

18 Mar, 2020

1. NSLS XFM HDF5 device:
 - a. Base initially on PNC-CAT HDF5 device as a starting point.
 - b. Extract hyperslice of HDF5 data-set for 'counts' to divide quasi list-mode stream into buffers based on Y index (varies fastest) and detector channel (slowest).
 - i. Use groups 'mca1', 'mca2', ... but ignore 'mcasum'. Note that 'mca8' is zero, which is OK using counts weighting for DT (see below).
 - ii. Get lost counts from 'inpcounts' – 'outcounts', and total counts from 'inpcounts' for each channel.
 - iii. Form 'dead_fraction' as ratio of total (across all detectors) lost to total counts, scaled by dwell time map (as assumed by 'da_evt', which will divide 'dead-fraction' by 'dwell' to arrive at DT fraction).
 - iv. Nominal 'dwell' estimated using a histogram approach, to not get biased by a few long-dwell pixels. Hence, this gets most common dwell, not the average.
 - c. 'Scalars/IO', etc are rates, not counts, so need to weight values by dwell time per pixel. Should use "raw" PVs now that DT is estimated and passed back.
 - i. Should probably veto all but 'raw' PVs.
 - d. Get positions from 'positions/pos' array. In case below, this can have a negative extent (final X less than initial) and hence a size that is negative in mm.
 - e. See "NSLS XFM HDF5 data.docx" doc in data/NSLS/XFM/Jan-2020.
2. EDS HMSA map device:
 - a. Convert BCF file to HMSA using BCF2HMSA program (~20x bigger).
 - b. Use 'xml_to_hash()' to read XML.
 - c. Read stripes in Y per read_buffer read (base on XFM HDF5 device).
 - d. No flux, charge or dwell as yet.
 - i. Should we get the pixel dwell as an input, and use Time 'flux' measure?
 - e. No absolute size either.
 - i. How to get pixel size in?
 - f. See doc "Reading HMSA and XML Data Files" in data/CSIRO/Bruker/.
3. OMDAQ device for Ljubljana
 - a. Cater for Header_Version 14 (rundata 9, adcinfo 9, file-structure 2) and LMF-version 4.
 - i. Add 'eqn' struct to sample in rundata
 - ii. Add QscalerCount to end of rundata.
 - iii. Expand adcinfo, and add 'timing' struct.
 - b. For LMF version GE to 4, also now read pixel address lookup table.
 - i. Event now has PixelCount and Composite E,channel,multiplicity Long word.
 - c. See doc "OM DAQ ListMode updated 3.docx" in GeoPIXE-local/OM/2020.

1 Feb, 2020

1. GeoPIXE
 - a. socket_command_set
 - i. One call to 'socket_command_set_item' missed the 'fail_on_retry' option.
 - b. Socket_command_set_item
 - i. Log_message any retry returns.
 - c. Warning
 - i. Include routine name in 'log_message' text.
 - d. Log_warning
 - i. Like warning, but only sends log_message.
 - e. Socket_retry
 - i. Sends a 'log_warning' to log attempt and success.
 - f. Save_image_all_HTML
 - i. Must not use the xmin/xmax, etc in call to 'make_tvb', which now do clip (as in 'image_save_all_HTML').
 - ii. Veto calling 'write_png', 'write_gif' if sx or sy is not 2 or larger.
 - iii. Test dimensions of b[] rather than sx, sy, in case of compression.

- g. Layer_plot
 - i. Layout:
 - 1. New X axis: Z
 - 2. New Y axis: Escape depth
 - ii. Events:
 - 1. Y – if 5, then set Z for X.
 - 2. X – if 2, then set Escape depth for Y.
 - iii. Plot_yields:
 - 1. If $(*(pstate).p1).microns$, then shows as microns, else mg/cm²
- h. Version 7.5v
- 2. Scan_list
 - a. Scan_list_maia_scan
 - i. Log_warning before only 'socket_command_set' of 'newrun'.
 - b. Version 7.0
- 3. OM DAQ device:
 - a. Start to implement header version 14, which uses LMF version 4 and Rundata (dataSTRUCT) 9 and ADCinfo 9.
 - b. See "OM DAQ ListMode update3.docx" in "GeoPIXE-local/OM/2020".

20-30 Jan, 2020

- 1. Generic device
 - a. Get_ascii_spec
 - i. Need to define DevObj as a valid "GENERIC_DEVICE", unique for each spectrum pointer
 - ii. Else, 'write_spec' throws an error.
- 2. Socket_command_set
 - a. If /next (do_script:)
 - i. If 'blog.newrun' then set 'fail_on_retry'.
 - ii. If 'socket_command_set_item' returns error and /fail_on_retry, then return with error=1 set to abort from script sequence.
- 3. Maia-setup
 - a. In timer loop (on 'pulser-start')
 - i. If 'socket_command_set, /next' returns error, then do not restart timer to abort from script sequence.
- 4. Maia_client_parameters
 - a. Bracket all requests for 'status.blog'fs.size', etc. with $((*pm).control.status.blog \text{ eq } 1)$
- 5. GeoPIXE:
 - a. Corr:
 - i. Added new menus to Association window to be able to further refine an element-element selection (GeoPIXE 7.5u). Now you analyze the spline selection using "Include points within spline" menu. Then on different element axes, you can further trim down this selection using two menus: (i) "further refine within spline": using a new spline on the new two-element plot, only the points from the first selection that also are contained within the new spline are kept, and (ii) "reject selection within spline": use new spline as a tool to edit the selected points, deleting any points within the new spline curve.
 - ii. Added a check-box (next to Z scale droplist) to enable display of current pixel selection, highlighted in green. This shows the current selection, after any multiple 2D field 'include' or any 'exclude' edits.
 - b. Corr_eventcb
 - i. Corr_analyze_spline
 - 1. New keywords: 'fresh', 'exclude', pass these top 'make_corr_mask'.
 - c. Corr_routines
 - i. Make_corr_mask:

1. New keywords: 'fresh', 'exclude'. Fresh=1 for a new spline include, else further refine current selection using new spline, and exclude=1, which uses new spline to edit/delete points from current selection.
2. NOTE: Need to distinguish 'q' index into Assoc 2D plane from 'qc' index into vectors of pixel conc for each element.
- d. Flux_flatten:
 - i. Change minimum base from 0.01 to 0.001
- e. Version 7.5u
6. Plugins
 - a. Flatten_to_flux0 plugin
 - i. Base default changed from 0.05 to 0.01.

9 Jan, 2020

1. Maia-device:
 - a. Added 'no_change_options' parameter to 'self'.
 - b. Instead of using 'no_change_options' on call to 'update_header_info' method, use a new method 'change_options' to set this mode in 'setup' method.
 - c. Now 'update_header_info' uses 'self.no_change_options' instead.
2. GeoPIXE:
 - a. Image_table_eventcb:
 - i. Image_Table_EVT:
 1. Bracket call to 'get_header_info', used to get YLUT table with "change_options, 0" and "change_options, 1" to stop the flip.x mode being set in 'update_header_info' method.
 - b. Fit_setup
 - i. Tweaked layout dimensions in Linux mode.
 - c. Version 7.5t

6 December, 2019

1. Maia-control
 - a. Maia_client_parameters_slow:
 - i. Get error on dimensions of "V" mismatch, so improve error testing for the following: (*pm).channel.trim.E.b (and E.a,T.a,T,b), data.index, (*pm).DA.bname, rGamma, deadtime.cal, scan.X,Y, scan.origin, scan.origin.bXunit, scan.pitch.bXunit, ..., scan.pitch.
2. Fortran (image_lib.f):
 - A. iXRF_events1
 - I. Add hack that checks for a "DP" and skips until it finds it. Should get removed when we get other records of interest, not just "DP".
 - B. Lib version 40

5 December, 2019

1. Fortran (image_lib.f):
 - a. iXRF_events1
 - i. Catch skip negative and set BAD=1.
 - ii. Treat BAD as ERR, return idebug=i where "DP" should have been.

25 October, 2019

1. DAQ Control
 - a. DAQ-launch
 - i. Pass on defaults from DAQ_device to DAQ_rates.
 - b. DAQ_rates
 - i. Add keyword for default.
 - ii. Add label text from conf file to display of detector # with rate.
 - c. Stage_list:

- i. Add extra delays before testing for raster (to move to origin) done.
 - ii. Print statements to monitor state, position.
 - iii. Remember to use /string, /float to “get” these types.
 - iv. Version 3.2
 - d. Version 7.5s
- 2. DAQ 36 device
 - a. DAQ_defaults
 - i. “label3”, etc. to label NIM channels 32, 33, 34, 35 on Rates map.
- 3. GeoPIXE:
 - a. Xrf_calc_int
 - i. Only correct ‘branch’ in f[z] if c30 (L), c50 (M) are not zero.
 - ii. Now we can fit Pb M as well as L with continuum source.
 - iii. This will require yields that need to fit M lines, using continuum source, to be recalculated.
 - b. File_search2
 - i. Is skip or cancel then abort search.
 - c. Plot_image_select
 - i. Only show info pop-up once per session.
 - d. Open_comms
 - i. Cannot use this (using Linux python 2.7.15+):
 - 1. Lclass.MMlog.LOG_LEVEL = Lclass.MMlog.INFO
 - ii. Must use this:
 - 1. Lclass.MMlog.LOG_LEVEL = Lclass.INFO
 - e. Pixe_fit
 - i. /tweak_par to adjust some slider values (after a fit).
 - f. Fit_setup
 - i. Use /tweak_par on ‘pixe_fit’ call for adjusting sliders.
 - g. Pixe_setup
 - i. If /tweak_par and ‘na’ defined, then do nothing.
 - ii. (makes use of a, org, rorg, mask, ... in common)
 - h. Pixe_initial
 - i. If /tweak_par, then don’t change a[] values for peaks.
 - i. Source_setup
 - i. Fixed display (was x1000) of overlay.
 - j. Detector_mimic
 - i. Adjust size of label text for DAQ rates display (legend=1, lstyle=1).
 - k. Image_table_eventcb
 - i. OnButton_Image_Table_Star
 - 1. Add new option to “Update region #0 pixel selection to include only pixels from other regions”.
 - 2. Add new option for “Delete the selected region and its matching spectrum”.
 - a. But ‘spectrum_select’ MUST be open for it to work.
 - ii. OnButton_Image_Table_Hotspot_Update
 - 1. Update region #0 pixel selection to include only pixels from other regions.
 - iii. new_table_pselect
 - 1. If no ‘i’ arg, then just use (*pstate).sel and set pointer for Notify.
 - l. Image_eventcb:
 - i. image_Results_Table
 - 1. Add return Notify flag ‘image-region-delete’.
 - m. Spectrum_display:
 - i. Add TLB Notify flag ‘image-region-delete’.
 - n. Spectrum_select:
 - i. Add TLB Notify flag ‘image-region-delete’.
 - o. Batch_sort
 - i. Test for Image_process and image_RGB windows open at ‘start’ now.

- ii. Add 'load' flag to pbatch struct, to indicate that old DAI file needs to be loaded, else do not load it (in Sort EVT 'start-evt' Notify code).
- p. Version 7.5s

24 October, 2019

1. GeoPIXE:
 - a. XOS_transmission
 - i. Scale by $\exp(-0.19 \cdot (E > 10.) - 10.)$ to better track source spectrum and pure element standards (see doc: "**Continuum yields tests.docx**").
 - ii. This falls off a bit faster than $1/E^4$ at higher energies.
 - iii. But Mo is the one exception, falling about 50% high (Zr, Ag, Sn OK).
 - iv. **NOTE: This requires source spectra to be recalculated – save with “-poly2” in the name.**
 - v. Yields will also need to be recalculated using these new source models:
 1. Load the “poly2” source version.
 2. Generate yields (add “poly2” to name, unless fixed std format).
 3. Save in new LCM file (add “poly2” to name, unless fixed std format).
 - b. Select_element_lines
 - i. For the 'continuum' source, only add Compton lines for source lines, not Elastic. Use normal element fitting for these.
 - c. Fit_setup
 - i. Increased span of spread to 15 (use about 3.3 for MM).
 - ii. Increased shift to -0.3.
 - iii. Now can fit Ag on wider Compton peaks.
 - d. Version 7.5r
2. Scan_list:
 - a. Including the new 'open_comms' change (setting LOG... attributes to the MMlog class instead of the MMcomms class) was needed, it seems.
 - b. Version 6.6

11 October, 2019

1. GeoPIXE:
 - a. Geopixe_defaults():
 - i. Parse “logging server” for rsyslog error log server.
 - ii. If not present, use the host for KVS.
 - iii. Fixed bug in logging server default on KVS endpoint line.
 - b. Strip_file_keys
 - i. Handle case of no extension, or file ending in “.”
 - c. Spectrum_load_plugins
 - i. Make sure error=1 unless good return struct.
 - d. Open_comms
 - i. Need to import MMlog (as Lclass) separately to Comms class, and then set LOG... attributes on Lclass.
 - ii. Python 2.7.13, 2.7.15+ both fail unless this is done.
 - iii. Seems both are more 'lazy' at loading implicit modules.
 - e. Version 7.5q
2. Maia_launch:
 - a. Init:
 - i. Use “prefs.logging.server” for error log server.
 - ii. Log prefix, Maia identity and GUI startup banner.
3. Scan_list:
 - a. Init:
 - i. Use “prefs.logging.server” for error log server.

10 October, 2019

1. Maia_launch:
 - a. Init:
 - i. If KVS enabled in 'geopixe.conf' then setup comms object
 - ii. Save 'comms' in pstate to use for log_message
 - iii. Save 'kvs', 'kvs_prefix' in pstate, although not used yet.
 - iv. Also copy to logging_comms in common for Warning and Alarm_popup implicit error logging.
 - b. Events, timers:
 - i. Implicit 'log_message' via all Warning and Alarm_popups.
 - c. Maia_rates
 - i. Add a clear button, to clear red max histogram data
 - d. Spawn_maia_clients, spawn_blog_clients:
 - i. Add server to args for Linux for rsyslog error logging in clients
 - ii. For Windows, add line but comment server out, due to Python issues on Windows (crazy Anaconda enviroments and poor env passing).
2. Maia, blog clients
 - b. Maia, Maia slow, Activity, DA2, Epics, ET2, Group spectra.
 - c. If argv[2] present, use as 'server' to open 'log_message' channel to rsyslog.
 - d. Log any read error timeout, rest, exit, as well as any Warnings (implicit).
3. GeoPIXE:
 - a. Warning
 - i. Added implicit 'log_message' if logging_comms in common valid
 - ii. Common: c_logging_comms, logging_comms
 - b. Alarm_popup
 - i. Added implicit 'log_message' if logging_comms in common valid
 - ii. Common: c_logging_comms, logging_comms
 - c. Open_comms
 - i. save_comms_in_common
 1. New routine to save comms to logging_comms in common for Warning and Alarm_popup implicit error logging
 - ii. NOTE: All programs in an IDLDE session will use the same log file, named by the first program to run (after a .f).
 - d. Version 7.5p
4. Scan_list:
 - a. No 'log_message' near warnings, as these are now implicit.
 - b. Version 6.5

7 October, 2019

1. Scan_list:
 - a. Init
 - i. Pass 'server' to open_comms for logging.
 - ii. Get 'server' from 'kvs_endpoint'.
 - b. Timer loop
 - i. Use 'log_message' to log any errors.
 - c. Scan_list_pause
 - i. Also pause Maia using 'scan_list_maia_pause' and close shutter using 'scan_list_shutter_pause'.
 - ii. Use new 'req_vsub_persist' to wait and loop and try again.
 - iii. Use 'log_message' to log errors.
 - iv. Needs GeoPIXE version 7.5o.
 - d. Scan_list_maia_pause
 - i. Use 'log_message' to log errors.
 - ii. Routine to pause Maia using "event.enable=0".

- e. Scan_list_shutter_pause
 - i. Use 'log_message' to log errors.
 - ii. Routine to close shutter during a pause.
 - f. Scan_list_start
 - i. Use 'log_message' to log phases and errors.
 - g. Scan_list_next
 - i. Use 'req_vsub_persist' for move to origin and raster execute.
 - ii. Use 'log_message' to log phases and errors.
 - iii. Needs GeoPIXE version 7.5o.
 - h. Version 6.4
2. GeoPIXE:
- a. Open_vsub
 - i. Req_vsub
 - 1. Added timeout as a keyword.
 - 2. Default timeout = 10 s now.
 - 3. Added '/notimeout' keyword to wait. But may not be a good thing to use, as daemon/ZMQ may have died/restarted and lost request.
 - ii. Req_vsub_persist
 - 1. Do 'req_vsub' and on error wait and loop and try again.
 - 2. Pops up a Warning with timeout and cancel option.
 - 3. Uses 'log_message' when error or timeout occurs.
 - iii. Get_vsub_server_active
 - 1. Use method 'vsub.is_usable()' instead of 'vsub.server_active'
 - b. Open_comms
 - i. Init
 - 1. Sets up MMlog for logging to mr-04-mel.
 - 2. Pass node as parameter "server", use for ztap too.
 - ii. Log_message
 - 1. Uses logger defined in 'open_comms' to log to rsyslog.
 - c. File_requester
 - i. Clarify explicit read or write versus none specified, which now defaults to write but shows as "Select File" with an "OK" button.
 - d. Version 7.5o

20 September, 2019

- 1. GeoPIXE:
 - a. Load_image_table_table, Onbutton_Image_table_export
 - i. Use where() to exclude all 'Back' and 'Sum' columns.
 - b. MPDA and multi-back?
 - i. Added code to 'read_da' to re-normalize all back rows to charge (@9-19)? Is this correct?
 - c. Pixe_fit
 - i. Add 'back_split' struct "{mode:0, emid: 10.0}" to results.
 - d. Fit_results:
 - i. Write back-split to PFR file
 - ii. New version -15
 - iii. Add parameters to output table.
 - e. Read_fit_results
 - i. Read new version -15 for 'back_split'
 - f. Fit_setup
 - i. Load/save PCM with 'background2', 'back2_split_energy'
 - ii. New version -12
 - g. Split_back
 - i. Make default espan = emid/8 (was emid/5).
 - h. Append_da_fit

- i. Veto optimization of back scaling for now.
- i. Version 7.5m

10 September, 2019

1. GeoPIXE:

- a. Read_fit_results
 - i. Bug: Set atweeks array to 10 elements NOT n_pars.
 - ii. Now able to use 'org', 'rorg' not just one of (10,0), (20,10).
- b. Pixe_fit
 - i. See lines here and elsewhere tagged "@9-19"
 - ii. 'atweeks' should only have length 10, not n_pars
 - iii. 'pback' now assumed to be pointer to an array of 2 pointers, each to spec structs for the two part background. 2nd one zero unless 2 pointers valid.
 - iv. 'background2' pulled from secod (non-zero) pback struct.
 - v. Now have 'background1', 'background2' for parts and 'background' for sum of these.
 - vi. Define 'back2_non_zero' logical when 2nd back present.
 - vii. Pass 'background1', 'background2' to 'pixe'.
 - viii. **Have NOT fixed results.background yet.**
- c. Calc_da_loop
 - i. Use "q = where(mask eq 1, nq)", after call to 'calc_da_matrix2', in case 'mask' has changed.
 - ii. Detect both mask[7]=1 and mask[10]=1 to apply test matrix multiply.
 - iii. Use 'name[]' not 'el[]' – was a bug.
- d. calc_da_matrix2
 - i. Split out 'background1' and 'background2', as in 'pixe_fit'.
 - ii. Define 'back2_non_zero' logical when 2nd back present.
 - iii. Enable mask[10] when 'back2_non_zero'
 - iv. When 'back2_non_zero' and 'back_mask' then add 2 additions for back to 'cmdl', 'rGamma', 'counts_per_ppm_uc'.
 - v. Tidy up use of 'back_offset' to be consistent with gamma and all back options.
 - vi. Pass 'background1', 'background2' to 'pixe'.
 - vii. Take care defining 'pq1' to explicitly set pq1 = -2 for mask[7] and pq1 = -1 for mask[10].
 - viii. Detect both 'back_mask' (mask[7]=1) and 'back2_non_zero' (mask[10]=1) to set 'el' and 'mu_zero' vectors.
- e. Fit_results
 - i. nc_max = maximum columns (for A selections)
 - ii. nc = nc_max (for A selections), but rows may have a different length ('nck')
 - iii. Now can read/write PFR files with mixed ORG,RORG align.
 - iv. Added 'Back2' tab with back2_mode and back2_split_energy to control the splitting of the background in two – these also added to the *p parameters struct.
 - v. (*pstate).pback now points to an array of 2 pointers, each to spec structs for the two part background.
 - vi. If 2nd one is zero, then it is ignored in plots.
- f. Pixe_initial, pixe_cleanup
 - i. Only need sum 'background'
- g. Pixe
 - i. Needs both 'background1' and 'background2'.
- h. Image_Process_Suppress_Region
 - i. Handles single pixel regions now.
 - ii. Does not do smoothing of region edges is $w < 1.0e-6$, for speed.
- i. Pointer_display
 - i. Shows values if array[1] encountered
- j. Layer_setup
 - i. Error setting detector mode, did not set from event.index.

- k. Version 7.5L
- 2. Combine & Align image plugin
 - a. Now handles write/read of shifts and a Zero shift option.
 - b. Works with MM Appended runs (zero offset needed).
- 3. Falconx_device
 - a. Falconx_listmode.def
 - i. spatial1_length = 74 for now.
 - ii. spatial1_rate_mask = '0007FFFF'x1 (was '0003FFFF'x1).
- 4. Scan list
 - a. Event loop – timer ‘start-button’
 - i. If error on ‘scan_list_retry_kvs’, then set timer again.
 - b. Scan_list_retry_kvs
 - i. On hard error, set err=1 and clear lock_retry_kvs.
 - ii. If fail 10 times, pop-up 10s warning, and try again.
 - iii. Use “Cancel” to drop out of loop.
 - c. Version 6.2

9 September, 2019

- 1. GeoPIXE:
 - a. Layer_setup
 - i. Array check box and array droplist used incorrect value for (*p).array.
- 2. Scan_list:
 - a. Read ‘append’ text before doing Append function.
 - b. Version 6.1

22 August, 2019

- 1. GeoPIXE:
 - a. Detector_select
 - i. Fixed bug in ‘array-mode’ set droplist.
 - b. Identify2
 - i. Use 3 lines for Help under Linux.
 - ii. Image_routines
 - 1. Make_tvb
 - a. Set minimum dilate for annotations/ circles to “ceil(ytgt/500)” so they always remain visible.

21 August, 2019

- 1. GeoPIXE:
 - a. Version 7.5j
- 2. Fortran:
 - a. iXRF_events1:
 - i. Scaled flux by flux_scale.
- 3. iXRF device
 - a. iXRF_device__define:
 - i. Fix flux_ic parameters in read_setup

5-20 August, 2019

- 1. GeoPIXE:
 - a. Image table export
 - i. Add rX,rY relative coordinates columns to the hot-spot export.
 - b. imageRGB_eventcb
 - i. Need to make sure droplist items are unique, just for **Linux**.
 - ii. Do this by appending “ 2” to the element names for 2nd list.
 - c. Version 7.5i
- 2. FalconX device

- Cleaned up definitions of changes with versions (0.8.1, 0.8.7, 0.9.0) as in def file: "falconx_listmode.def".
- Separated out AS-version part.
- This also affected fx_browse.

[illegible]

3. iXRF device

- a. `iXRF_device__define`:
 - i. Based on Midas device as a template.
 - ii. Uses DT and weight based on dwell.
 - iii. Perhaps should use estimated counts for multipass merge (as in DAQ device)?
 - iv. Add flux-scan stuff, and default name, scale from header.
- b. `iXRF_accumulate_dtfx`
 - i. Not using `flux_scale` yet? Should it?
- c. `Read_ixrf_json`
 - i. Hacked to allow 16-bit (multi-char) strings in JSON.
 - ii. Will need to correct for UTF-8 later.
- d. `Read_ixrf_header`
 - i. Just uses `'read_ixrf_json'` and `'get_iXRF_details'`.
- e. `get_iXRF_details`
 - i. Assumes `IC_sensitivity = 1.0` and `IC_name = 'iXRF:dwell.time'`
 - ii. Takes Cal from header, which may not be correct.
 - iii. Assumes `energy = KV` for now.

4. Fortran:

- a. iXRF_events1:
 - i. Assumes 16-bit “DP” characters for now. Will need to fix when all strings become UTF-8.
- b. falconx_events4:
 - i. Uses 48 bit sample, as in events3.
 - ii. Pass ‘tick’ to Fortran, passed via common from header read.
 - iii. Cleaned up definitions of changes with versions (0.8.1, 0.8.7, 0.9.0) as in def file: “falconx_listmode.def”.
 - iv. Separated out AS-version part.
- c. falconx_events3:
 - i. Allow for dwell (sample count) to be 48 bit (2x 24 bit LSB, MSB).
 - ii. spatial1_sample_msb_mask = z'00FFFFFF'

1. `dwell_last = tcount * tick`
 2. `tcount = iand(tev, spatial1_sample_msb_mask)`
 3. `dwell_last = dwell_last + tcount * tick * 2.**spatial1_sample_msb_offset`
 - iii. NOTE: 'tick' still fixed (4.0e-6)
 1. Needs to be passed ...
 2. (based on 'afe.sampleRate' found in 'falconx_version')
 - iv. Lib version 39
5. Scan-List
- a. If X,Y origin changed before an append (or after followed by an apply) then set the coords to match the source coords (e.g. Finder).
 - i. This assumes that any change to origin is in stage/Ruler coords.
 - ii. Then ready for a "Translate".
 - iii. Added a changed test to both 'scan_edit_read', and 'scan_edit_update, /from_panel' as they are a bit different. Need to resolve some duplication here sometime ...
 - b. Version 6.0

3 August, 2019

1. Maia_launch:
 - a. At startup, use `maia_defaults()` to set obj Flip.X parameter.
 - b. Also pass FlipX to DA process in `ppar[9]`.
 - c. Version 7.5h
2. Blog_client_da2
 - a. Pick up FlipX in `ppar[9]`.
 - b. Flip using: `x = nx-x` (may leave a gap on the left side).
 - c. This probably buggers up the X origin in saved RT image. This will need some work ...
3. GeoPIXE:
 - a. Image table EVT

2 August, 2019

1. GeoPIXE:
 - a. Image table EVT
 - i. In non-cluster mode, the call to 'spec_evt' is missing pileup, throttle and linearize parameters. It then defaults to those in header, which may be old (original behaviour). Now added.
 - b. Version 7.5h

1 August, 2019

1. Maia_control
 - a. Only restore GeoPIXE.sav if called as 'maia_control', not 'maia_launch'. This enables debug under IDL 8.7.2 (as maia_launch) WITHOUT loading a GeoPIXE.sav compiled under IDL 8.5.1 (with bad python).
 - b. This follows the lead of 'mm_scan_list' versus 'scan_list'.
 - c. Version 7.5g
2. GeoPIXE:
 - a. Open_socket
 - i. If error code returned from 'socket', change error=1 before return from 'open_socket'
 - b. Socket_retry
 - i. If the 'open_socket' fails, now pop-up a warning (with timeout=10) and then try opening the socket again. Cancel to abort.
 - ii. Honors maximum retries parameter, if not zero.
 - c. Translate_region
 - i. If no points are found in current image space, then return `q = -1` and `err=0`. Else it aborts an 'update-all' loop.
 - d. Analyze_image
 - i. Include region # (i_update) in no points error pop-up.

- e. OnNotify_Image
 - i. 'image-region-update-one' notify edit to align with update-all.
- f. Image_table hotspot neighbourhood
 - i. Increased exclusion around hotspot using kernel s2.
 - ii. Remove an exclusion zone around ALL hotspots (q0) from each final neighbourhood.
- g. Image RGB Learn – Execute
 - i. Popup options for desired zoom level.
- h. Options_popup
 - i. Allow new /exclusive option for check-boxes.
 - ii. Then initial value is just a scalar index.
 - iii. Return for 'r.check' remains a vector (w/ only one set)
- 3. Scan_list
 - a. To acknowledge changes in GeoPIXE libs (e.g. sockets), change version.
 - b. Version 5.9

30 July, 2019

- 1. GeoPIXE:
 - a. Image_table
 - i. Work on default file-names for Load and Save.
 - ii. Make last row selected after a load, as you get a 'images-results' notify anyway, which forces last row.
 - b. detector_update
 - i. Looks for detector files in geopixe, but also 'config/detectors', with config defined in geopixe.conf.
 - ii. Use 'file_search(dir, filter)' to look in subdirs too.
 - iii. Use 'sort_unique(/last)' to sort and make config duplicate take precedence.
 - c. filter_update
 - i. Looks for filter files in geopixe, but also 'config/filters, with config defined in geopixe.conf.
 - ii. Use 'file_search(dir, filter)' to look in subdirs too.
 - iii. Use 'sort_unique(/last)' to sort and make config duplicate take precedence.
 - d. Fit_setup
 - i. Used 'detector_update' and 'filter_update' extensively.
 - ii. These probably also need to update drop-list at some time.
 - e. Layer_setup:
 - i. Used 'detector_update' in detector drop-list, load LCM and realize.
 - ii. Use detector_update in /array mode only.
 - f. Sim_setup_geopixe
 - i. Used 'detector_update' and 'filter_update' extensively.
 - g. Depth_ratio_yield
 - i. Used 'detector_update' in load yields.
 - h. Detector_select
 - i. Used 'detector_update' in droplist.
 - i. identify2
 - i. Used 'detector_update' and 'filter_update' on droplists.
 - j. Result_properties
 - i. Used 'detector_update' and 'filter_update' extensively.
 - k. Image_eventcb:
 - i. image_Save_all_HTML
 - 1. Now need to call make_tvb without min/max parameters, which now do a clip.
 - l. Version 7.5g
- 2. Wizard_depth
 - a. Used 'detector_update' in load yields.
- 3. Wizard_standards
 - a. Used 'detector_update' in load yields.

- b. Version 2.4

29 July, 2019

- 1. GeoPIXE:
 - a. Spec_evt:
 - i. Cluster mode, new mask approach to not blow memory. New yrange3, dyoffset3 (offset beyond initial 'yoffset') defined.
 - ii. In cluster mode, need to call 'get_ylut' method to populate YLUT internally for later 'range_ylut' call.
 - 1. This is bad.
 - 2. Really need a way to flag 'header not read yet'.
 - iii. Xrange, yrange passed to device_specific must be full scan range for the Flip to work properly (not just original_xsize, etc.).
 - iv. Build a new mask for just the stripe of data in this process, with indices order for Y slowest.
 - v. Use new 'spec_accumulate4' and 'spec_det_accumulate4', which has new mask order and a 'yoff' to offset from yoffset to start of mask Y.
 - b. Spec_accumulate4
 - i. New Fortran routine with new mask, plus yoff.
 - c. Version 7.5f

25 July, 2019

- 1. GeoPIXE:
 - a. Batch_sort
 - i. Only sets 'save' if Image Operations returns a do-it flag.
 - ii. Likewise 'overwrite'.
 - b. Image_operations
 - i. On 'batch-filter' notify, only set a do-it return flag if a valid operation was performed.
 - c. EVT:
 - i. On 'start-evt' notify, set *p comment, sample, grain from the 'head' header values from raw data file.
 - d. File_requester
 - i. Use new 'file_search2' to search down dir trees, but with the ability to cancel out. Cancel jumps straight to the file-requester popup.
 - ii. First tries the local path and the remote path, without a dir tree search.
 - e. File_search2
 - i. Like 'file_search' using separate 'dir' and wildcard 'spec', but pops up a progress bar, which has a 'cancel' button to abort.
 - f. Version 7.5e
- 2. FalconX device
 - a. Update_header_info::
 - i. Allow sample_info_1, etc. metadata – concatenate into sample_info.
 - ii. Find missing beam_energy
 - iii. Add sample, grain, comment

22 July, 2019

- 1. GeoPIXE:
 - a. Image_rgb
 - i. Now receives the 'image-analyze-mark' notify to redraw with moved shape.
 - b. Plot_images, plot_RGB_images
 - i. Catch error on west if window closed and set flag zero, to open a new one.
 - c. Image_routines
 - i. Make_tvb
 - 1. Operate on just a sub-region of bc array to do fiddly circle tests.
 - d. Image_RGB_routines

- i. Make_RGB_tvb
 - 1. Operate on just a sub-region of bc array to do fiddly circle tests.

18 July, 2019

- 1. GeoPIXE:
 - a. Image_routines
 - i. Make_tvb
 - 1. Check (sy > 400) if compress is not less than 1, not sy*compress, and then dilate highlights for no-screen case.
 - b. Image_RGB_routines
 - i. Make_RGB_tvb
 - 1. Check (sy > 400) if compress is not less than 1, not sy*compress, and then dilate highlights for no-screen case.

17 July, 2019

- 1. GeoPIXE:
 - a. Strip_file_keys
 - i. Strips out selected key strings, element names, file_M terms, etc. to simplify th construction of files names with additions.
 - b. Image_eventcb, image_RGB_eventcb, image_table_eventcb
 - i. OnButton_Image_Table_HotSpots, OnButton_Image_Table_HotSpot_Neighbourhood, Export
 - 1. use strip_file_keys
 - c. plot_image_select, plot_RGB_image_select
 - i. Save plot options as version -4
 - ii. Load plot options and then set widgets update.

16 July, 2019

- 1. GeoPIXE:
 - a. Set_device
 - i. 'Z' device
 - 1. Added 'Z'. Pass 'aspect' set from Preview window aspect.
 - 2. Use 'scale_z' = 10 to pass to 'default_plot'.
 - 3. Use device 'screen' of 15000 pixels across
 - 4. Scale_z = 10
 - 5. If /true set_pixel_depth=24 else set_pixel_depth=8
 - ii. 'METAFILE'
 - 1. Increase device screeen size to 15000 / !d.x_px_cm
 - 2. Adjust scale_wmf = 11
 - b. Default_plot
 - i. Add 'Z', similar to Win, but scaled * scale_z
 - ii. Tweaked charsize for 'METAFILE', 'Z' and screen
 - c. Pcharsize
 - i. Add 'Z', similar to Win, byt up 20%.
 - ii. Doubled charsize scaling wmf_factor to 1.5.
 - d. Plot_images
 - i. Z device (for /PNG, /JPEG options)
 - 1. Use as screen=1, used_non_screen=0, (png=1 or jpeg=1)
 - 2. Get aspect ratio from preview window
 - 3. if /PNG
 - a. Save red, green, blue from 'tvltc'.
 - b. At end, do tvrd() and write_png
 - 4. if /JPEG
 - a. Must open 'set_device, /true' to use 24 bit colours and then tvrd(/true) and write_jpeg, /true at the end.

- e. Image
 - i. Add menus for plot PNG, JPEG
- f. Image_eventcb
 - i. Image_export
 - 1. Add /PNG, JPEG option
- g. Plot_image_select
 - i. Add PNG, JPEG to droplist
- h. Image_RGB_eventcb, plot_RGB_images, plot_RGB_image_select
 - i. Also add /JPEG following PNG.

15 July, 2019

1. GeoPIXE:
 - a. Batch_sort
 - i. Notify 'batch-operations-open' and 'batch-rgb-open' when options[0] and [11] are set to tell image window to open the operations and RGB windows, if they are not open.
 - ii. Use the new routine 'Set_table_select' to set active table row.
 - b. Image_rgb
 - i. Notify 'image-rgb-closed' when closed to image, which maintains a flag 'RGB_open', like for image_operations.
 - c. Image_eventcb
 - i. On 'batch-rgb-open' notify, open RGB window if not open (RGB_open).
 - ii. On 'batch-operations-open' notify, open image_operations window if not open (operations_open).
 - iii. Pass the Notify flags 'batch-operations-open' and 'batch-rgb-open' up when open EVT.
 - d. Image_table
 - i. Load_image_table_table
 1. Add 'Pixels' column
 - ii. OnButton_Image_Table_Export
 1. Add /hotspots option, hs_el selection.
 2. If /hotspots get XY centroid from mode 5,6, add to output.
 - iii. OnButton_Image_Table_Star
 1. Option 4 calls: OnButton_Image_Table_Export with /hotspot
 - e. EVT
 - i. Pass the Notify flags 'batch-operations-open' and 'batch-rgb-open' up when open batch_sort.

14 July, 2019

1. GeoPIXE:
 - a. Image_table
 - i. Revise starting file-name on save. Don't strip path.
 - b. Evt_start
 - i. Use updir=4 to find raw data
 - c. Evt:
 - i. Add updir=4 for evt_button load, and updir=3 for pileup, throttle, linearize file requester loads.
 - d. File_requester
 - i. Use 'fix_path()' on translate table paths before comparisons.
 - e. Version 7.5d
2. Maia device
 - a. Update_header_info::
 - i. Do not check for energy = 24.1 to test for FlipX axis, just test facility and endstation.
 - b. Read_setup::
 - i. Problem with using 'read_maia_32_header' based on 'unit' as this will NOT be blog .0 in parallel processes.
 - ii. Now uses file-name from fstat(unit) and calls 'get_maia_32_header'.

- c. Get_maia_32_header:
 - i. Strips off ext and forces blog file .0 for header read.

12 July, 2019

- 1. GeoPIXE:
 - a. Batch_sort
 - i. Notify 'done-evt':
 - 1. Notify 'batch-filter' even if no filter needed, passing new .'do_it' in pargs0.
 - 2. Do not test for digital filter here anymore.
 - ii. Notify 'done-filter':
 - 1. Only Notify 'batch-save' from here now.
 - b. Image_eventcb
 - i. OnNotify_Image
 - 1. 'batch-filter': check for process window open, else open it.
 - 2. 'image-process': do the processing.
 - ii. Image_save
 - 1. Was error in 'ext' if /overwrite, fixed.
 - c. Image_process
 - i. OnNotify_Image_Process
 - 1. 'batch-filter': Only call 'OnButton_Image_Process_Get' when do_it true. Notify 'done-filter'.
 - ii. OnButton_Image_Process_Get
 - 1. Read only once and store in new (*pstate).pmodel.
 - 2. Notify, 'image-process' passing list.

9 July, 2019

- 1. GeoPIXE:
 - a. Image_table_eventcb
 - i. OnButton_Image_Table_Delete
 - 1. Now can delete multiple rows after drag select (not row label column).
 - ii. OnCellSelect_Image_Table
 - 1. Only select whole line if click on row label, else just set (*pstate).sel.
 - iii. OnButton_Image_Table_Load
 - 1. On load set (*pstate).adc to -1, so we default to "array" on the EVT droplist.
 - iv. Image_table_EVT
 - 1. Use updir=3 to search for raw data file.
 - b. Image_eventcb
 - i. Image_analyze
 - 1. Pass through /get_stats option.
 - ii. OnNotify_Image
 - 1. Wizard – 'sum-region' now passes (*pd).get_stats on to 'image_analyze'
 - 2. Must use Standards Wizard 2.3 or later.
 - c. Image_routines
 - i. Analyze_image
 - 1. New /get_stats option to calculate sd, rels, else don't.
 - 2. Only used/called from Standards Wizard.
 - d. Corr, corr_eventcb (Association):
 - i. Make log Z (Z axis=2), X,Y smooth = 1, X,Y linear axes the defaults.
 - e. Round_kernel(w)
 - i. Function returns a bytarr kernel to use with Dilate.
 - ii. Use this with Dilate() in 'make_tvb', 'make_RGB_tvb' and hot-spots.
 - f. File_requester:
 - i. At each updir step, also 'file_search' down from there for a match to the file-name.
 - g. EVT:
 - i. Evt_set_evt_file

1. Use /no_output to veto setting a new output filename based on input file-name.
- h. Evt_start:
 - i. If blog file not found and go searching for it, call 'evt_set_evt_file' with /no_output to NOT set a new output file-name and /no_cal_adopt to NOT ask for yes/no on E cal and pileup/throttle.
 - ii. Use strip_path() on 'cluster_client' title.
- i. Parallel_client
 - i. Add 'pending' return from 'parallel_monitor' to show number of precces remaining.
- j. Cluster_client
 - i. Use 'pending' return from 'parallel_monitor' to show number of precces remaining in sub-title.
 - ii. Put 'cluster_client_update' after 'parallel_monitor' in event routine.
 - iii. 'cluster_client_update' adds pending to sub-title text.
- k. Version 7.5c
2. Standards Wizard
 - a. Use new /get_stats (passed in pdata in Notify) to force calculation of sd, relsd in Analyze_image.
 - b. Plot now works if only one point.
 - c. Version 2.3

9 July, 2019

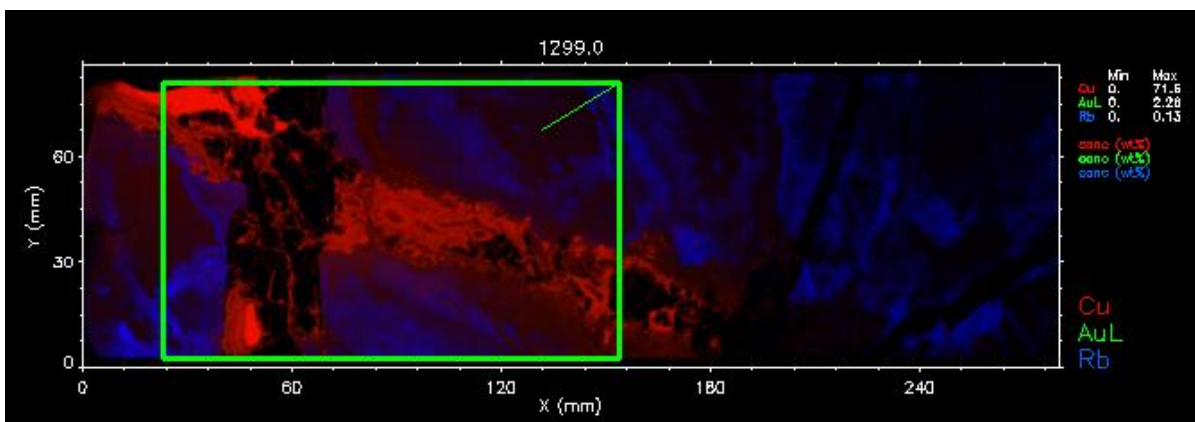
1. GeoPIXE:
 - a. Image_table_eventcb
 - i. Table view
 1. Throughout, using the new 'set_table_select'.
 - ii. OnButton_Image_Table_Star
 1. Now uses 'options_popup' to select between options for hot-spots, hot-spot neighbourhoods, deleting all regions, except 0, kill pixels from all regions in all image planes.
 - iii. OnButton_Image_Table_HotSpots
 1. Now forms a useful (*pstate).file name for saving.
 - iv. OnButton_Image_Table_HotSpot_Neighbourhood
 1. Uses Dilate to sample the neighbourhood of each hot-spot.
 2. Forms a useful (*pstate).file name for saving.
 - v. OnCellSelect_Image_Table2
 1. The meat of 'OnCellSelect_Image_Table' that can be used to select a cell in the table, with flow-on Notifications.
 - vi. OnNotify_Image_Table
 1. 'image-results', 'image-region-throttle',
 - a. Uses new 'widget_control, (*pstate).table, set_table_select'
 - vii. OnButton_Image_Table_Delete:
 1. Does not set table select, as the notify select will.
 - viii. OnButton_Image_Table_Delete_all:
 1. Delete all but first (if /keep)
 2. Veto sending 'image-results' notify (???) as this just comes back to image-table anyway.
 - ix. OnButton_Image_Table_Clear
 1. Veto the no_regions in 'image_table_event'.
 - x. OnButton_Image_Table_Delete
 1. Veto the no_regions in 'image_table_event'.
 2. If n=1 just call clear.
 - xi. OnButton_Image_Table_Star
 1. Use options_popup to gather parameters/ options

2. Implement hotspots and selective delete, clear pixels in all planes for all regions.
3. Notify, 'image-kill-regions-all-planes' for the latter.
- xii. OnButton_Image_Table_HotSpot_Neighbourhood
 1. Samples pixels around hot-spots
- b. Image_eventcb
 - i. OnNotify_Image
 1. 'image-kill-regions-all-planes':
 - a. Loops through regions and kill pixels in all planes.
 - b. Uses new 'image_region_select', with kvs=0
 2. 'image-region-select':
 - a. Uses new 'image_region_select', with 'kvs
 3. 'image-region-update':
 - a. Now gets elx, ely from (*p).elx, ...
 - ii. Image_Process_Kill_All_Region
 1. Calls 'Image_Process_Suppress_Region' in a loop to suppress to zero each region in all element planes.
 - iii. Image menu:
 1. Process→clear→Kill all regions in all planes
 2. Calls 'Image_Process_Kill_All_Region' to kill all.
- c. Image_routines
 - i. image_region_select
 1. Select region 'i' and display shape
 2. Optional /kvs to also set KVS for Box shape
- d. set_table_select
 - i. Manipulate widget_control for the table so that you select a row(s), keeping it in view, but without too much jumping around, keeping row 0 visible, for example.
- e. Options_popup
 - i. Added 'initial_check' input.
 - ii. Changed default sizes.
- f. Plot_images, plot_RGB_images:
 - i. Use Catch to use an existing window, 0, if present.
 - ii. This enables one to be resized and used for new preview.

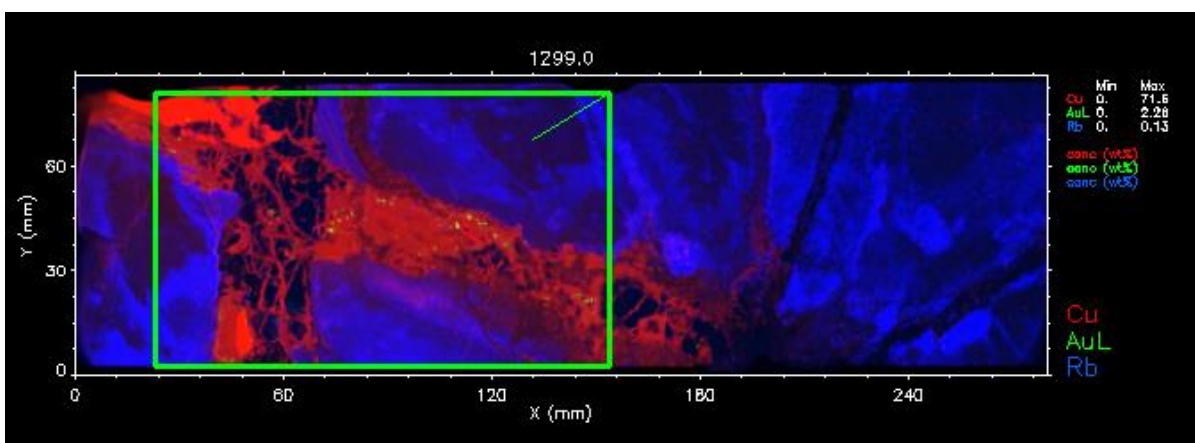
5 July, 2019

1. GeoPIXE:
 - a. Image_routines
 - i. Make_TVB
 1. Removed redundant 'crop' option. There was an option to use xmin,xmax, etc. which is done better.
 2. Revised how the /nozoom options work to modify 'compress'
 3. This allows growth as well as shrinkage.
 4. Use dilate on bc in both the nozoom and normal operations. Dilation factor based on either compress<1 or for large display size.
 5. Nozoom=0 option now test for compress<>1 instead.
 6. Show centroids as circles in bc[].
 - b. Plot_images
 - i. Crop now passed to Make_TVB xmin/xmax, etc. parameters using the /nozoom option.
 - ii. In fact /nozoom used throughout because the Make_TVB output 'b' is not intended for the screen buffer dimensions.
 - iii. Added centroids option and an element for X,Y
 - iv. Image scaling in TV output device used now, not by congrid.
 - v. 'Plot_mark' needed to use the zoom() function for scale calculation after the other changes.

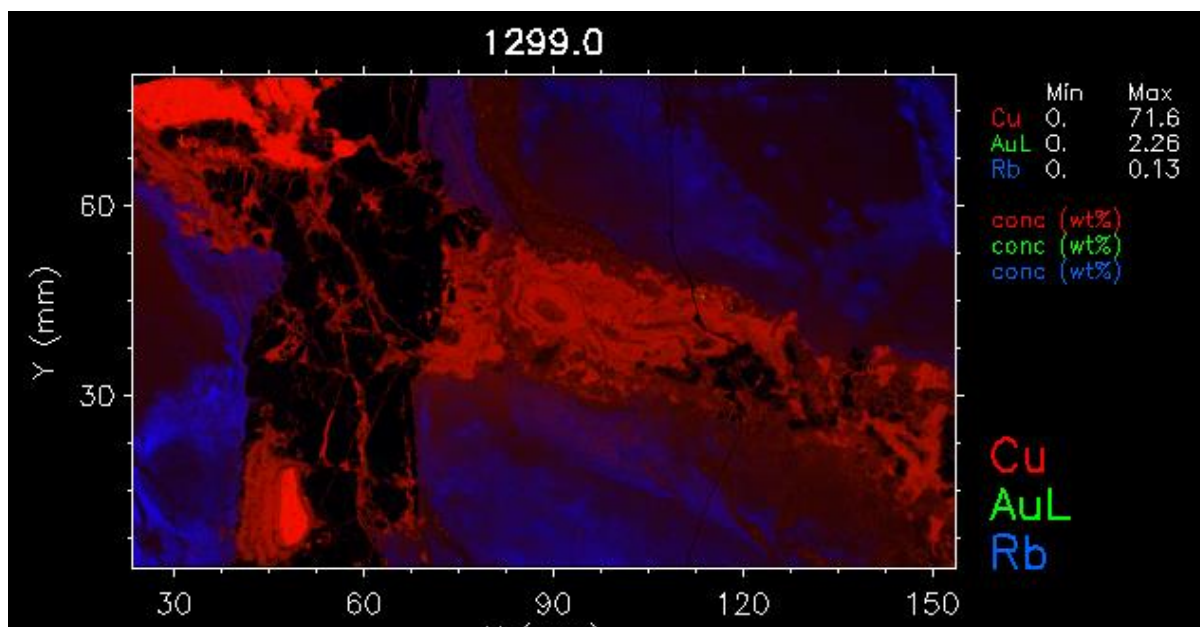
- c. Image_RGB_routines
 - i. Make_RGB_true
 - 1. New options for nozoom, xmin,xmax, ymin,ymax, xtgt,ytgt, enhance, which all silently get passed to make_RB_tvb.
 - ii. Make_RGB_tvb
 - 1. Now uses xmin,xmax, ymin,ymax (with /nozoom) instead of 'crop'.
 - 2. Added new 'enhance.spots' option to do a dilation on each plane. Dilation based on either compression factor or for large display size.
- Plot_RGB_images
 - iii. Crop now passed to Make_TVB xmin/xmax, etc. parameters using the /nozoom option.
 - iv. In fact /nozoom used throughout because the Make_TVB output 'b' is not intended for the screen buffer dimensions.
 - v. Image scaling in TV output device used now, not by congrid.
 - vi. Added new 'enhance spots' option to do a dilation on all planes. This tends to 'enliven' colour images ... (see below). But this will also conceal/ lose small holes/ local minima in images.
 - vii. Added a field to select which element planes to enhance.
 - viii. Fixed image index and opt references for "compare" images.
- d. Define
 - i. New /plot_options
 - 1. Use it in: plot_spectrum_select, plot_rgb_image_select, plot_image_select, plot_corr_select.
 - 2. Referenced in: plot_spectra, plot_images, plot_RGB_images, plot_corr.
- e. New routines to load/save plot-options: 'save_plot_options' and 'load_plot_options'.
 - i. New version -4



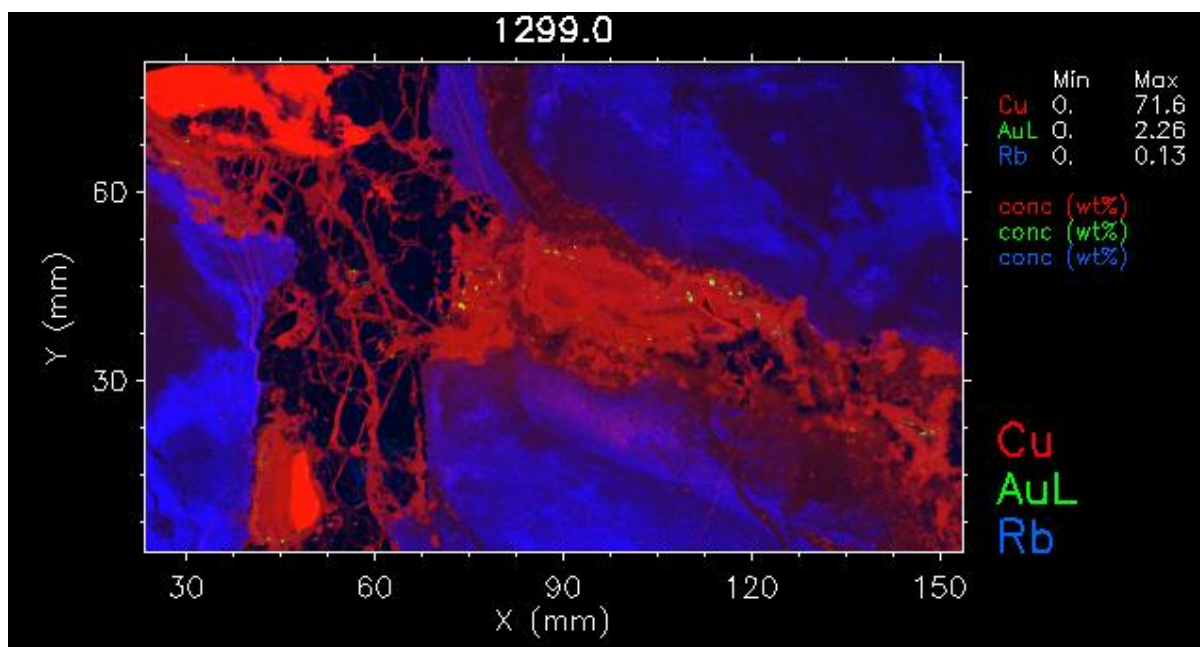
Normal RGB export (no "enhance spots")



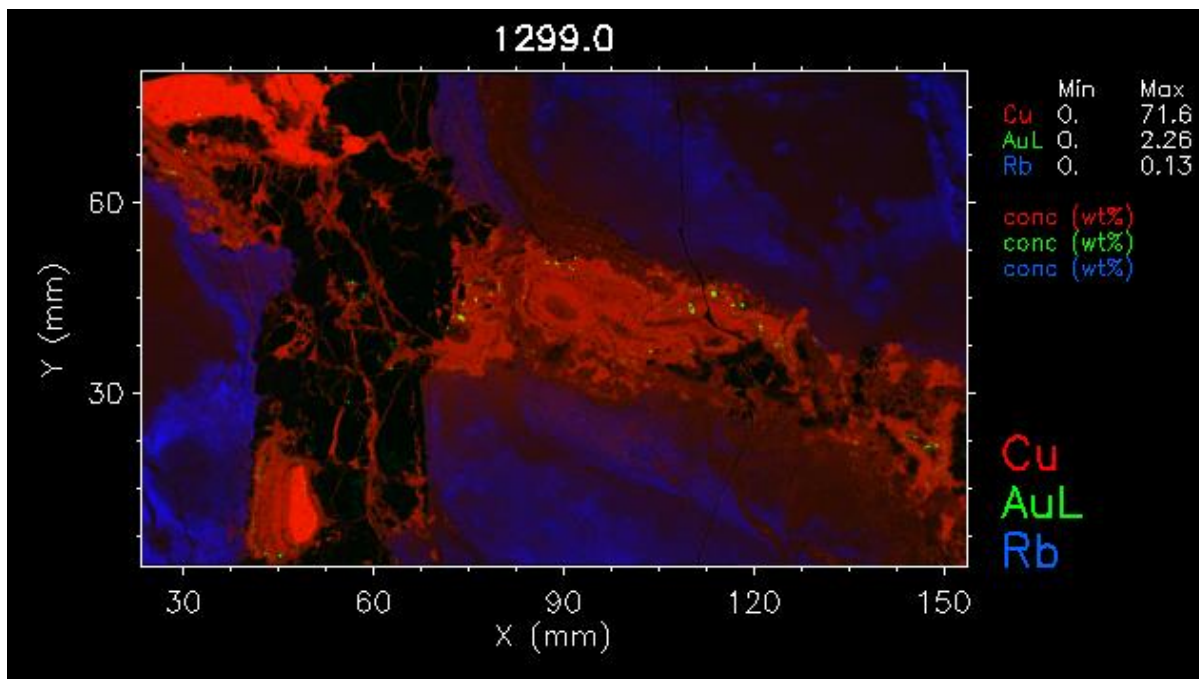
New RGB export ("enhance spots" set), which tends to also enhance weak other elements (?)



Normal RGB export (no "enhance spots")



New RGB export ("enhance spots" set), applied to all planes



New RGB export ("enhance spots" set), applied to just Au plane

3 July, 2019

1. GeoPIXE:
 - a. Image_absolute
 - i. Clarify that 'crop' is in compressed units, while the normal 'x_sub_range' is not.
 - b. Image_routines
 - i. Make_tvb
 1. Use 'dilate' function to expand the *qc array 'bc' to make sure single pixel hot-spots remain visible when zoom < 0.
 - c. Version 7.5b

2 July, 2019

1. GeoPIXE:
 - a. Image_absolute
 - i. Function to determine all display viewport parameters (org, size) in absolute and relative coords, pixel coords, uncompressed pixel coords, with optional 'crop' input for a selected region.
 - ii. Copes with original image, windowed sub-range, and shape region applied.
 - iii. Deals with FlipX, FlipY device mirroring, in which case 'origin' is the right corner (X) or top corner (Y) for absolute cords, pixel cords, but not uncompressed, which is used for windowed sort setting.
 - iv. Determines the absolute coordinates of origin and size for the displayed area, or an optional sub-region, specified via 'crop' in pixel units. Make use of device FlipX (or FlipY) to indicate a flipped axis.
 - v. crop = {x:[min,max], y:[min,max]} will accept single point for x,y.
 - vi. Returns r = {absolute:abs, pixel:pixel, uncompressed: uncompressed}
 1. Absolute: absolute stage coordinates of origin,size of image (or crop if provided) (mm).
 2. Pixel: pixel coordinates of origin,size of image (or crop if provided).
 3. Uncompressed: pixel coordinates of origin,size of image (or crop if provided) in uncompressed units.
 4. Range: Range for XY that can be used in PLOT for xrange, yrange (Flip will flip axis plot).
 - vii. If FlipX, then absolute origin and pixel origin will be on the right corner. If FlipY, then absolute origin and pixel origin will be on the top corner. However,

uncompressed origin will ALWAYS be the bottom-left corner for windowed sort usage.

- b. Onbutton_image
 - i. Uses 'image_absolute' to determine region/shape and scan parameters.
 - ii. Mods to 'UP' for Box (1), Circle (2) and S.Point (11).
- c. Image_details
 - i. Fix bug in showing absolute origin, with or without a sub-region selection, with axes with a FlipX or FlipY device attribute.
 - ii. Use new 'image_absolute' routine for window and cropped dimensions and XY range setting for plot.
- d. Plot_images
 - i. Use new 'image_absolute' routine for window and cropped dimensions and XY range setting for plot.
- e. Make_tvb
 - i. If crop is used, then then *qc overlay needs to be applied to full 'bc' array before taking a crop subset of 'bc'.
- f. Plot_RGB_images
 - i. Use new 'image_absolute' routine for window and cropped dimensions and XY range setting for plot.
- g. Test case: MM run 925 in Demo:
 - i. Full image (no compress):

```
Data file(s):
/OSM/PER/CESRE_1160-MDU/work/Demo/MM/blog/925/925.0 ...
Maia 384 (Sync) - HYMOD data acquisition
Valid: 7036469812, bad XY: 0, clipped: 11720725
Facility: MM.Mel., Endstation: 1
Pileup: /OSM/PER/CESRE_1160-MDU/work/Demo/MM/config/384D19/pileup/90
Size:
2628 x 756 pixels
Compression: 1 x 1
Original: 2628 x 756 pixels (1.9 M pixels)
Size: 78.84 (mm) x 22.68 (mm)
Origin: 387.015 (mm) x 62.825 (mm)
1. Bounds X: 0 to 2627, Y: 0 to 755
```

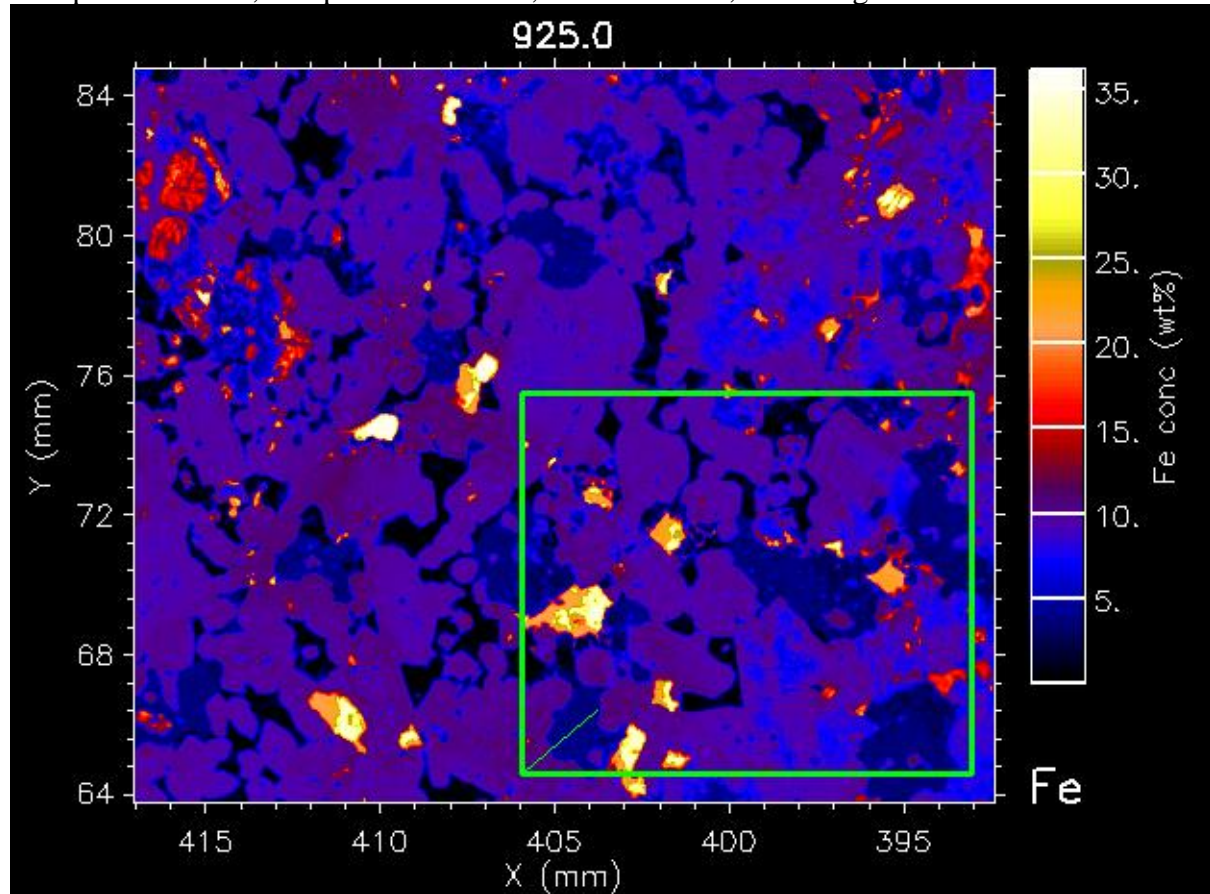
- ii. Compressed full image (2x2):

```
Data file(s):
C:\Software\Demo\MM\blog\925\925.0 ...
Maia 384 (Sync) - HYMOD data acquisition
Valid: 7036469812, bad XY: 0, clipped: 11720725
Facility: MM.Mel., Endstation: 1
Pileup: C:\Software\Demo\MM\config\384D19\pileup\900-wide.pileup.var
Size:
1314 x 378 pixels
Compression: 2 x 2
Original: 2628 x 756 pixels (1.9 M pixels)
Size: 78.84 (mm) x 22.68 (mm)
Origin: 387.015 (mm) x 62.825 (mm)
1. Bounds X: 0 to 1313, Y: 0 to 377
```

- iii. Windowed sort "Win1" (created from 2x2 compressed):

Data file(s):
 C:\Software\Demo\MM\blog\925\925.0 ...
 Maia 384 (Sync) - HYMOD data acquisition
 Valid: 2189839906, bad XY: 0, clipped: 4398021824
 Facility: MM.Mel., Endstation: 1
 Pileup: C:\Software\Demo\MM\config\384D19\pileup\900-wide.pileup.var
 Size:
 822 x 700 pixels
 Compression: 1 x 1
 Sub-region:
 Sub-range: 822 x 700 (uncompressed pixels)
 Size: 24.66 x 21. (mm)
 Offset: 1626 x 32 (uncompressed pixels)
 New origin: 392.415 x 63.785 (mm)
 Original: 2628 x 756 pixels (1.9 M pixels)
 Size: 78.84 (mm) x 22.68 (mm)
 Origin: 387.015 (mm) x 62.825 (mm)
 Bounds X: 1626 to 2447, Y: 32 to 735

- 1.
- iv. Compared full sort, compressed 2x2 sort, and Win1 sort, and all agree.



Fe image (925) for Win1 with a further small region/shape displayed 'absolute'

1 July, 2019

1. GeoPIXE:
 - a. Version 7.5a
2. Scan_list:
 - a. Timer changes:
 - i. Timer_sequence - 1.5 → 2.5 (larger than coords)
 1. Does this stop the occasional "Frozen" warning popup?
 2. Also may help get a good/changed stage status return.
 - ii. Timer_coords – keep this at 1.2 s
 1. Now also check for 'status.blog.discard'
 - b. Alarm popups
 - i. Stage frozen should be tighter or polled slower – timer should help.
 - ii. Detect raster on, but Maia not logging (discard=1).
 - c. Version 5.8b

3. Maia_launch:
 - a. Alarm popups
 - i. Maia_launch_zerorate:
 1. If (*pm).run.rate drops below 1 B/s, flag warning popup.
 - b. Version 7.5a.

26 June, 2019

1. GeoPIXE:
 - a. Image_routines
 - i. Analyze_image
 1. Better treat negative pixels in calculation of XY centroid positions of regions.
 2. If total goes negative or zero, then clip at +1, which forces an average over all pixels selected.
 - b. Spectrum_select:
 - i. Whenever: notify '(*pstate).update_notify', pass (*pstate).pshow as pointer.
 1. Do this for: "R" button, Next, Prev, One, All, various delete channel options and 'array-select' receive Notify from 'detector_select'.
 - ii. NOTE: pselect is not for passing Notify data; it's the select window pars struct array, which lives in the parent.
 - iii. Open 'detector_select' window passing watch=(*pstate).update_notify
 - iv. Add a new heap pchannel to pass list of detectors to 'detector_select'.
 - c. Detector_select:
 - i. New 'watch' keyword to pass variable Notify name string, save in (*pstate).watch.
 - ii. On Notify 'watch', check pointer for a valid 'pchannel' channel enable vector.
 1. If found find matching detector widget address numbers (not detector channel numbers; use (*play).ref) and enable these.
 2. If not found, then it's a normal spectrum update, with no channel change, so ignore.
 - d. Plot_image_select
 - i. Fixed typos in struct assign on Load – strange?
 - ii. Fixed default file name if 'file' blank.
 - e. Plot_images
 - i. If image crop selected, do not plot shape.
 - ii. Fix flaw in METADATA device that has wrong character Y size.
 - iii. For long X aspect, reduce character size and line thickness.
 - f. ImageRGB
 - i. Overlay current shape on RGB image.
 - ii. No auto plot updates for now (needs double buffering, which uses a lot of memory).
 - iii. Add 'pstate_parent' to pstate to access parent image pstate parameters for crop.
 - g. ImageRGB_routines
 - i. make_RGB_tvb, make_RGB_true:
 1. Add crop option
 - h. ImageRGB_eventcb
 - i. ImageRGB_export – new to plot RGB images
 - i. Plot_RGB_image_select
 - i. Plot setup for RGB image plot, like for image window.
 - ii. Disable 'ppm only'
 - j. Plot_RGB_images
 - i. Plot RGB images in box, optional crop, like for image window.
 - ii. Display Min/Max and axis 'style' for Z legend
 - iii. Fix flaw in METADATA device that has wrong character Y size.
 - iv. For long X aspect, reduce character size and line thickness.
 - k. Load_spec_colours
 - i. Change 'd.blue' to look less dark on black.
 - l. Socket_command_set

- i. New option /fail_on_retry, forces an error return after a (single) socket retry.
 - m. Compiled using IDL 8.5.1: “Build GeoPIXE source 3 using IDL 8-5-1”.
- 2. Scan_list
 - a. Scan_list_start
 - i. Add a ‘scan_list_maia_stop’ at finish, which only occurs on error.
 - ii. Normal phase=3 return now sets phase=0 and jumps to preturn.
 - b. Scan_list_next
 - i. Add a ‘scan_list_maia_stop’ at finish (only for phase GE 2 where Maia has started), which only occurs on error.
 - ii. Normal phase=3 return now sets phase=0 and jumps to preturn.
 - c. Scan_list_maia_scan
 - i. Use the option /fail_on_retry in ‘socket_command_get’ of “newrun” to avoid sending lots of newruns in the event of some network/ NFS problem.
 - d. Compiled using IDL 8.5.1: “Build GeoPIXE source 3 using IDL 8-5-1”.
 - e. Version 5.8a

25 June, 2019

- 1. Versions and IDL 8.5.1 versus 8.7.2
 - a. Need to cope with the following:
 - i. Python bridge on Windows now needs IDL 8.7 due to Anaconda problems.
 - ii. Python bridge on Linux requires GeoPIXE compile to match IDL there.
 - 1. Python bridge only used for Maia Mapper.
 - iii. Normal GeoPIXE (no Maia Mapper) does not care.
 - 1. KVS (python) is used for Maia Mapper data in GeoPIXE.
 - b. Strategy:
 - i. Linux VMs
 - 1. Need to keep GeoPIXE, etc. compiled using IDL 8.5.1, until IDL 8.7 installed.
 - 2. Then Python bridge demands compiling for 8.7 when using IDL 8.7.
 - ii. Windows PC
 - 1. Now must use IDL 8.7 source and compiled for Python to work.
 - iii. FTP site
 - 1. 7.4:
 - a. That remains at 7.4@, compiled under IDL 8.5.1.
 - 2. Beta:
 - a. Use 7.5 (source3): must remain **compiled using IDL 8.5.1** for now.
 - 3. Later (no external will need Python):
 - a. Move beta (8.5.1) to new “7.5” dir.
 - b. Make “beta” a 8.7.2 compiled dir.
 - c. **Do this when CSIRO upgrades IDL to 8.7.2 internally.**
 - iv. Windows PC scripts:
 - 1. IDLDE 2 GeoPIXE source 2
 - a. Runs IDL 8.5.1 on GeoPIXE-source2 (now frozen)
 - b. Compile using IDL 8.5.1
 - 2. IDLDE 3 GeoPIXE source 3 No Compile
 - a. Runs IDL 8.7.2 on GeoPIXE-source3
 - b. **Do not compile from here** for now. Run all 8.7 in IDLDE.
 - 3. Build GeoPIXE source 3 for IDL 8-5-1
 - a. Runs IDL 8.5.1 on GeoPIXE-source3
 - b. **Compile here for FTP site and Linux** for now.
 - c. Anaconda issues
 - i. Murray found in the source, that a different approach to Tornado IO loops came in with version 17 of pyzmq around Dec 2017. Need to revert pyzmq to 16.0.2, which is what Murray is using.
 - ii. See more details in the Anaconda installation notes (C:\Install\python\Anaconda).

Murray will need to change code approach to Tornado before we move to pyzmq 17.0 and beyond.**24 June, 2019**

1. Organization
 - a. Fix references to GeoPIXE-source2 → 3 and also release/GeoPIXE2 → 3 in project properties and save_routines for geopixe, Maia launch, MM Camera View, MM Scan List.
2. GeoPIXE
 - a. Lmgr2
 - i. Use activate.exe to find install_num, site_notice when run from IDL 8.7.
 - b. Image_table
 - i. Increased table size to 50 x 10000 rows. Added 'rows_max', to pstate.
 - c. Version 7.5
 - d. Use folder /beta on FTP site for now.
3. Execution of IDL 8.7.2 sav file on Linux under IDL 8.5.1
 - a. A newer SAV file read into IDL 8.5.1 under Linux fails to load PYTHON::INIT.
 - i. This causes errors under Linux IDL 8.5.1.
 - b. NOTE: Do not try to use DL 8.7.2 version of GeoPIXE on Linux under IDL 8.5.1.
 - i. At least with kvs enabled in 'geopixe.conf'.
 - c. Cannot have IDL 8.7.2 version of GeoPIXE on Linux used with Python.
 - i. Not until IDL 8.7.2 installed under Linux.
 - d. Continue to use IDL 8.5.1 for builds for now.

22 June, 2019

2. GeoPIXE-source2 copied to GeoPIXE-source3
 - a. GeoPIXE-source2 will remain in IDL 8.5.1 mode.
 - b. GeoPIXE-source3 will now use IDL 8.7.

12 June, 2019

3. GeoPIXE:
 - a. Detector_show
 - i. Also offset the detector by thick/2 in Z (axis 2) so that the front face is at the desired distance.
 - ii. Added the local detector pad tilt to the oModel_pad[] object.
 - iii. Offset pad label -thick/2 also so it's not buried "inside" pad.
 - iv. Change Y offset for character to -0.35*pad_char_size.
4. Scan_list:
 - a. If scan start fails (error), then do a 'scan_list_maia_stop' to endrun.
 - b. Version 5.6a

23 May, 2019

1. GeoPIXE:
 - a. PCA routines
 - i. Tweaked display ranges, settings.
 - b. Bouncing windows (Sort EVT, Fit-setup, Correct_yields)
 - i. Found that the 'bounce' happens with "widget_control, update=1".
 - ii. Also occasionally setting a base widget (not TLB).
 - iii. Fixed using a new scheme:
 1. Bracket rendering using 'widget_control_update, tlb, update=0' and 'widget_control_update, tlb, update=1' (which includes resetting the XY offset). These can be nested.
 2. For overall rendering, which may include 'file_requester' popup, bracket using 'widget_control_update, tlb, /save' and 'widget_control_update, tlb, /rest'. The can only be used once, not nested!
 - c. Evt:

- i. Use a 'widget_control_update' /save, /rest bracket in template load, which has many widget changes mixed with file_requesters sometimes.
 - ii. Just use a simple 'widget_control_update' bracket in 'step_size_drop', 'image_mode'.
 - iii. Leave a few commented out, and the old 'widget_control' update code commented with " ;#".
- d. Fit_setup:
 - i. Just use a simple 'widget_control_update' bracket in 'load_pcm_parameters' and 'tab-panel'.
- e. File_requester:
 - i. Just use a simple 'widget_control_update' bracket in tree build in 'OnRealize_File_Requester_tree'.
 - ii. Use the simple bracket for all events that need to update/ grow the tree.
- f. Correct_yield:
 - i. Add simple update bracket to 'Load' and 'Import'.
 - ii. Fixed setting Y size of Rest and Original buttons.
- g. widget_control_update:
 - i. New routine to manage rendering updates while holding TLB position.
 - ii. It saves the position from geometry on update=0 and restores position on update=1. Because rendering is suppressed, we don't see it jump.
- h. Stringify
 - i. Include List, Hash, OrderedHash, Dictionary
 - ii. Use these chars to signify: Hash ("), OrderedHash (%), Dictionary (\$).
- i. Unstringify:
 - i. Include List, Hash, OrderedHash, Dictionary
 - ii. Hash, OrderedHash, Dictionary all decode as these now.
 - iii. Change a vector to a List, if the types are not compatible.
 - iv. IDL cannot Transpose an array of structs – veto this!
- j. Pointer_display:
 - i. Handle Dictionary
 - ii. Print using "{", "{" for Hash, OrderedHash and Dictionary
- k. Vector_merge:
 - i. Form a vector (used in 'unstringify', store in 'string_packer').
 - ii. Upgrade a type in a vector to maintain precision.
 - iii. If not compatible types, upgrade both to Lists.
 - iv. Make string the highest rank in precision.
- l. Var_type:
 - i. Like 'size(x,/type)' but also distinguishes between List, Hash, OrderedHash, Dictionary, ObjRef using 'typename()'.
- m. Version 7.4@
- 2. Idl_query_geopixe:
 - a. Conditional execution on: float(!version.release) ge 8.6
 - b. Use lmgr2 for >8.6
 - c. make_idl_query:
 - i. Add resolve the functions: lmgr2, strsplit, strip_char.

9 May, 2019

- 1. GeoPIXE:
 - a. Use fix_path() in geopixe_library, startupp, find_device_object, idl_query
 - b. Fixed typo in old spec_xstep_evt (pmaski).
- 2. Pixels_legend:
 - a. New function to convert pixel # to 'human readable' form.
 - b. Add this to 'image_details' for history window.
- 3. Time_legend:
 - a. New function to convert time (s) to 'human readable' form.
- 4. Pca_cluster

- a. Make_pca_cluster_tvb
 - i. Use 'image_weighted_max' to suppress central tight blob.
- 5. image_weighted_max
 - a. Add 'scope' input to control divide-down factor limit.
- 6. Socket_command_set, socket_command_get:
 - a. Add the command 'scom' to the error popups.
 - b. Add a timeout=10 in call to 'waning' prior to socket retry.
 - c. Successful set or get resets socket 'attempts'.
- 7. Warning:
 - a. Added 'timeout' keyword for timeout in seconds
 - b. If timeout is used call 'warning_popup' instead.
- 8. warning_popup:
 - a. New pop-up that both prints and pops up a multi-line message.
 - b. Has a timeout to auto-dismiss and a 'cancel' button option.
- 9. Standards wizard:
 - a. Wizard_standards_update_plot:
 - i. Enclose x,y in [] in oplot, in case they are scalars (nq1=1).
- 10. Blog browse
 - a. Add decoding of the new MK marker event types.
 - b. Version 7.4z
- 11. Scan_list
 - a. Replace 'scan_list_time_string' with a new generic function 'Time_legend'.

1 May, 2019

- 1. Scan_list:
 - a. scan_list_maia_scan:
 - i. Now set 'input.chan[0]' to "dwell.time".
 - ii. Then direct flux chan #0 to input #0 'flux.chan[0].source' = "in0".
 - iii. Add test on (*pm).version. kandinski gt 8600
 - b. Version 5.5a.
- 2. GeoPIXE:
 - a. Decode_elam
 - i. Using Elam name for Leta ("Ln") instead of geopixe name "Leta". Fixed.
 - b. Make_database
 - i. Generate a new database, saved to "geopixe2.sav".
 - c. Xrf_calc_int
 - i. K,L cascade contribution terms
 - d. Image_details
 - i. Added index 165 for number of phases line.
 - e. Image
 - i. If KVS not enabled, call 'open_kvs("null")' to get all compiled in IDLDE mode.
 - ii. This is needed to avoid errors in 'set_kvs_box' on region draw.
 - f. Version 7.4z
- 3. Maia-Control
 - a. maia_launch_overloss
 - i. Added run, segment, rate to discard/link error loss pop-up.
 - b. Version 7.4z
- 4. Updated all code on mr-06-per
 - a. With python not working on PC, need to compile geopixe, scan_list on Linux.
 - b. Built save files on mr-06-per.
 - c. Copy geopixe.sav, mm_scan_list.sav back to PC.

9 Apr, 2019

- 1. Image_lib.f:
 - a. falconx_events3:

- i. Increased 'spatial1_lengthB' from 11 to 12, and added test of tag bits. If test fails, continue at line 11 as a different record.
 - ii. This enables variable spatial records (e.g. for multiple detector ICR, OCR).
 - iii. Then added ability to terminate a record if tags change, so then ...
 - iv. Increase 'spatial1_lengthB' to 12 plus 2* number of additional detectors (e.g. 'spatial1_lengthB' = 12+30 = 42 for 16 detectors in all).
 - v. Indeed, it should be simply large enough now to exhaust words for this tag.
 - vi. If OCR, ICR get split to accommodate large dwells at high rates, then we'll need even more. But make sure that we don't stop too soon (length should be generous).
 - vii. Have 'tick' ready for input, but not implemented yet.
- b. Lib Version 34
- c. Tested with run #100 (0.8.3 format), #44 (0.9.0 format) OK.
- 2. Falcon_listmode.def
 - a. Added a new test for version 900 or higher to increase length of spatial statistics to 12.
- 3. Fx_browse:
 - a. It is fundamentally limited to fixed length records, so will work with version 0.8.3 and 0.9.0 now, but not for a merged record if they get variable in length.
 - b. However, some code in 'read_falconx' will terminate a record if the tag changes, so it might just work as is! Need to test it.
 - c. If so, need to change length to 42.
 - d. Tested with run #100 (0.8.3 format), #44 (0.9.0 format) OK.
- 4. GeoPIXE:
 - a. Fit_setup:
 - i. Added "cancel" on bad energy calibration warning to skip all fits.
 - b. Pointer_display:
 - i. Added handling of Hash, OrderedHash, List data-types.
 - ii. Changed way pointers are handled using recursion earlier.

8 Mar, 2019

- 1. Scan_list:
 - a. Made the stroke (append-text) editable:
 - i. Added event routine to set (*pstate).scan_sequence.stroke.
 - b. Version 5.4

6 Feb, 2019

- 1. Geopixe:
 - a. Peak_Fit_Stats_spectrum_plugin:
 - i. Added a new control for the 'FWHM median test energy', defaulting to 270 eV now.

14 Jan, 2019

- 1. Geopixe
 - a. Geopixe_defaults:
 - i. Add "prefix2" and "endpoint2" to prefs struct.
 - b. Image:
 - i. Use both prefix, prefix2 from geopixe_defaults() as kvs, kvs2 in pstate.
 - c. Onbutton_image
 - i. "up" for circle, s.pixel: use kvs, kvs2 from pstate, according to 'facility'.
 - ii. "up" for box calls 'set_kvs_box'.
 - d. Image_routines: set_kvs_box:
 - i. Use kvs, kvs2 from pstate, according to 'facility'.
 - e. Version 7.4y

5 Dec, 2018

- 1. Scan List:
 - a. 'mm_scan_list' now assumes command-line arguments:

- i. kvs-endpoint, sl-config, <others>

28 Nov, 2018

1. GeoPIXE:
 - a. Check_kvs:
 - i. Write “test” to a key that increments: “test.idl.0”, “test.idl.1”, ...
 - b. Open_vsub:
 - i. Check existence of endpoints in KVS first.
 - c. Spectrum_display
 - i. Add Guassian function to Analyze Fit
 - d. Spectrum_Do_Curve_Fit:
 - i. Default to 1/y weights, unless ‘has_errors’ true, then use 1/sigma^2.
 - ii. Use ‘call_procedure’ to construct fit result curve.
 - iii. Notify ‘spectra-changed’ after adding FWHM to comment after fit, so that spectrum-select updates its table.
 - e. Spectrum_routines: curvefit_initial
 - i. Add setup for ‘gauss_function’.
 - f. Spectrum_select: Load_select_table
 - i. Adjust column widths
 - g. Gauss_function
 - i. Initial and Gaussian function to use with curvefit.
2. Scan List:
 - a. KVS problem, object not defined.
 - i. Ignore test of KVS as a valid PYTHON object at start of retry.
 - ii. After each ‘check_kvs’ do not test for PYTHON object, just retry.
 - b. Retry KVS:
 - i. Loop up to 10 times, then pop-up warning.
 - ii. Must complete all KVS, VSUB, else starts again.
 - iii. Print rather than Warning of errors to continue.
 - c. Next:
 - i. Now does not print warnings, just print.
 - ii. If Error then retry KVS, VSUBs.
 - d. Version 5.2

3 Nov, 2018

1. GeoPIXE:
 - a. Onbutton_image
 - i. Show absolute position in Help for “S Pixel”.
 - b. Set_image_minmax:
 - i. If an image has limited data (e.g. small stripe(s)), then minmax using a border may miss it.
 - ii. If nothing_remains=1 due to the border applied, then zero border and try again.
 - c. image_weighted_max:
 - i. If total is zero, return nothing_remains=1
2. Maia-control:
 - a. Disk space warning: do not use time, but warn for <0.5 TB, <0.1 TB free.
 - b. Frozen stage: narrow test to <0.01 mm.
 - c. Add better messages for all other ‘maia_launch_over...’ routines/ warnings.
3. Scan List:
 - a. Scan_list_translate:
 - i. Define logicals ‘mirror1’ and ‘mirror2’ to indicate whether endstation 1, 2 should be mirrored.
 - b. Scan_list_maia_scan, scan_list_maia_stop:
 - i. If /test or (*ps).open = 0, then quietly return.

- c. Shutter name for output control was wrong for Perth. Changed definition of 'shutter_name' in start-up.
- d. Version 5.1

18 Oct, 2018

1. GeoPIXE:

- a. Notify
 - i. In case a bad parameter is passed in the first call to 'Notify', define {NOTIFY} first using simple terms.
- b. Get_maia_details:
 - i. Added "context" to calls to unstringify.
- c. Read_DA:
 - i. Added "context" to calls to unstringify.
- d. Struct_to_hash:
 - i. Recursively convert a struct to hash.
 - ii. /lower - to force lower case keys, else default upper case.
 - 1. Upper was not working – set default to lower=0
 - iii. Add 'lower=lower' keyword to nested 'struct_to_hash()' too, which was causing nested "Raster" spec to get wrong case keys.
- e. Hash_to_struct:
 - i. Use idl_validname to guarantee good tag names.
 - ii. Change order in struct, to append further hash keys.
- f. State_button:
 - i. Subvert a right mouse button click (only in freeze button mode) to send an alt select. Use this in rates window.
- g. Open_vsub:
 - i. req_vsub:
 - 1. 'n_elements(r)' does not work, even if 'r' exists! @10-18
 - 2. Use 'isa(r)' instead.
- h. Open_kvs:
 - i. Set_kvs:
 - 1. Use 'struct_to_hash' to recursively convert struct to hash.
 - 2. Option for lower case hash keys, or default uppercase.
 - 3. NOTE: hash to struct to hash will not preserve mixed case!
- i. Harden_beam
 - i. Bug if slices varied, as they do, away from an interface or surface, which meant way under-estimating the attenuation.
 - ii. Only affects Maia Mapper/ continuum yields.
- j. Geo_yield2:
 - i. Report yields for Maia Mapper as counts/ms, i.e. scale by 1/1000.
 - ii. Now with XOS transmission, get "conv" of about 0.25.
- k. Image_table:
 - i. clear_image_table_table:
 - 1. Only clears table, not preions data now.
 - 2. Data needs to be cleared explicitly only when necessary.
 - ii. free_image_table_all:
 - 1. Clear preions data from Image-Table, when needed.
 - 2. Possibly duplicates 'free_image_regions' in 'image_routines'.
 - iii. OnNotify_Image_Table:
 - 1. Only clear table (free_table_entry) for region load and clear button, not for Notifies on 'image-results', 'image-region-throttle', 'image-results-clear'.
- l. Image_analyze:
 - i. Bug forming presults ate end of routine – presults not defined always, fixed.
- m. Image_details:

- i. Photon per second: for normal PIXE, SXRF, where yields are in counts/uC we use the charge/time times 6.242×10^{12} . For Maia Mapper, where yields are in counts/ms, charge equals 'conv' * total dwell (ms). Then we need the total model spectrum counts per second, scaled by 'conv' for ph/s.
 - n. Find_charge_val_unit_index:
 - i. An error returned incorrect Unit index. Do not use 7.4v, 7.4w.
 - o. Onbutton_image
 - i. Added KVS position write from Circle centre too.
 - p. Select_evt_files:
 - i. Add an “explanation” for the offset popup.
 - ii. Version 7.4x.
- 2. Maia-control:
 - a. Maia-rates:
 - i. On a ‘detector’ event, react to a alt select event (from state_button) and send a ‘detector-toggle’ event instead to be used by Maia-setup to toggle pad selection.
 - b. Maia-launch:
 - i. Launch Maia_rates registering for ‘detector-toggle’ notifies to be passed on to Maia-setup.
 - c. Maia-setup:
 - i. React to a ‘detector-toggle’ Notify to toggle pad selection.
 - ii. Maia-setup launched registering for ‘detector-toggle’ notifies.
- 3. Scan-list
 - a. Define variable ‘clayton_wrong_left_right’ to use wrong end-station assignment (1=Left, 2=Right) in Clayton ONLY. If this is zero, use correct assignment (2=Left, 1=Right).
 - b. Use conditional on ‘clayton_wrong_left_right’ and (isite eq 0) to define droplist and context help strings that mention “Left” or “Right”.
 - c. Reads Kandinski parameters from .config KVS, based on endpoints key.
 - d. Determines end-station number ‘n’ from config key numeral (e.g. ...SL.n).
 - e. Use new “left”, “Right” shutter names to REQ to safety damon.
 - f. Version 5.0
- 4. Wizard_standard:
 - a. Fixed error of PV added to table without fixing other table parameters.
 - b. Return pointers for presults kept getting lost.
 - i. Turned out that a Notify from Image in Image-Table was not just clearing the region table, but also clearing the pregions array.
 - ii. Now pregions only cleared when explicitly necessary.
 - iii. And the table clear just clears the table.
 - c. Needed a fix of Image-table updates elsewhere. The pointer data for regions cannot be cleared willy-nilly.
 - i. Only clear it ‘at source’, e.g. when loading images or regions, not at the end of a Notify.
 - d. Region done callback now calls “wizard_standards_find_match” to find the relevant standard value, instead of first with right element (bad)..
 - e. Version 2.2
- 5. Maia device:
 - a. Set flip.x True if energy = 24.1 (recent), and either:
 - i. strmid(facility,0,6) = “MM.Mel” and endstation = “1”
 - ii. strmid(facility,0,6) = “MM.Per” and endstation = “2”
 - b. Set flip.x False if energy = 24.1 (recent), and either:
 - i. strmid(facility,0,6) = “MM.Mel” and endstation = “2”
 - ii. strmid(facility,0,6) = “MM.Per” and endstation = “1”
 - c. Add ‘no_change_options’ keyword to ‘update_header_info’ method to prevent any changes to sort options called from ‘setup’ method.

9 Oct, 2018

1. GeoPIXE:
 - a. Notify
 - i. In case a bad parameter is passed the first call, define {NOTIFY} first using simple terms.
 - b. Get_maia_details:
 - i. Added “context” to calls to unstringify.
 - c. Read_DA:
 - i. Added “context” to calls to unstringify.
2. Maia Control:
 - a. Maia_launch:
 - i. Added “context” to calls to unstringify.
 - b. Maia_update_parameters2:
 - i. Added “context” to calls to unstringify.
3. Strings
 - a. Unstringify:
 - i. Added “context” keyword to error messages, so that the source can be seen better.
4. Scan-list
 - a. Added a MirrorX option to the transformation ‘t’.
 - i. Detects this for both IN and OUT coordinates.
 - b. Set this if end-station=1 (Left).
 - c. Default back to “Finder” on start-up.
 - d. Pass kvs_prefix to scan_edit.
 - e. Version 4.1
5. Scan-edit:
 - a. Added a droplist to select a default reference between “ruler” and “finder” and a get button to load these for current frame.
 - b. Added kvs_prefix to call to scan-edit and pstate.
6. Set_default_reference:
 - a. Routine to copy a selected frame reference to a default for ruler or finder.
 - b. Only used manually, standalone,
7. Get_default_reference:
 - a. Routine to retrieve a default frame reference for ruler or finder.

2 Oct, 2018

1. GeoPIXE:
 - a. Batch processing
 - i. Scan_batch_dir:
 1. Options popup to select run # range.
 - ii. Scan_dir_evt:
 1. Uses rmin, rmax range on run numbers.
 - iii. Batch_sort:
 1. On ‘done-filter’ did not parallel new path and TIFF settings in ‘pargs1’ as in on ‘done-evt’. Fixed.
 2. On Notify ‘root’ now prompts to ask to correct all paths to same rules, then corrects all paths using ‘build_output_path’.
 3. Only enabled rows are corrected.
 4. Delete DAI only works within the run-number range.
 5. Added another ‘option’ for saving RGB images.
 - iv. Sort EVT:
 1. ‘*(pstate).dpath’ was stale, needed to update from evt_file.
 2. Needed to Notify ‘dpath’ as well as ‘path’ and ‘root’ so that Batch Sort would be up to date.
 - v. Image RGB

1. Add the “Learn” menu to create, save, restore and execute a list of RGB displays at full resolution (zoom=0).
2. Act on ‘batch-save’ Notify to save a list of RGB combinations.
- vi. ImageRGB_Learn:
 1. New routine to to create, save, restore and execute a list of RGB displays at full resolution.
- vii. Image:
 1. Needed to register the ‘batch-save’ notify when opening RGB window.
- b. Charge_gain_unit_lists:
 - i. Add “ms” to units.
 - ii. Add ‘times’ list to flag which unit is a time.
- c. Charge_sensitivity:
 - i. New return arg “time”
 - ii. Detect “ms” as a unit, and return time=1
- d. Find_charge_val_unit_index:
 - i. Find a ‘unit’ in the list, used for drop-list index.
 - ii. Now use /time switch to restrict to time or not units.
 - iii. Change all occurrences in **EVT, flux_select, image_details, spectrum_details, maia_setup** to locate (l) ‘time’ in PV and then pass ‘time=(l ge 0)’.
- e. get_maia_details:
 - i. If ‘flux_chan0_unit’ is a time (“ms”) then set IC_name = “Maia_dwell.time”
 - ii. Detect sample name in blog header as “Udimet” and set sample_serial=”500” and sample_type = “standard”.
 - iii. HACK: for Udimet, set beam energy=24100.
 1. Only if ‘energy’ not set.
 2. A real synchrotron data-set should set a correct energy.
- f. Base device:
 - i. Add null methods for “flipX()” and “flipY()” that return false.
- g. Onbutton_image:
 - i. Use obj->flipX() (and obj->flipY()) to offset Box origin back onto original stage coordinates. This ensures that the Box (in mm) can be put back into stage space (new region scan, etc.).
 - ii. Pixel origin is kept unchanged in flipped pixel coordinates, to be used in Windowed sort “Get”, for example.

GeoPIXE version 7.4v

2. Scan List
 - a. hash_to_struct
 - i. New stage daemon returned a key "KEY" equal to !NULL sometimes for scan update, so added:
 1. if (d eq !NULL) then d=0, to stop error with ‘create_struct’
 - b. scan_list_maia_scan:
 - i. If sample name is “Udimet” or “Udimet ...”, then set sample.serial to “500” and sample.type to “standard”.
 - ii. If sample name is “NIST” or “NIST ...”, then set sample.serial and sample.type to “standard”.
 - iii. Need a better way to flag a “standard” sample.
 1. Use ‘metadata.sample.type’ in future.
 - iv. Set beam energy to “24100” for Maia Mapper for now.
 - v. Set flux.chan[0] ‘name’=”dwell.time”, ‘unit’=”ms”, ‘coeff’=1.0.
 1. (note that at sync these are set by beamline controls)
 - vi. Set ‘metadata.scan.dwell’ to nominal dwell time (s).
 - c. Scan List version 3.3
3. Wizard_standards
 - a. Scan_dir:
 - i. If gain is zero from get_header_info, then set gain=1 ???

1. This is needed for 'Maia: dwell.time' mode, it appears.
 - ii. Add "pv" to row struct, and set this from 'mp.IC_name'.
- b. Process_blog:
 - i. Set 'flux_scaler' to (*p)[j].pv, which should now be "Maia: dwell.time".
4. Maia device:
 - a. Add "Flip X" and "Flip Y" check-box to 'sort_options' and rendering.
 - b. Increase Y size to 213 (was 190).

13 Sep, 2018

1. define:
 - a. Added 'stroke' to 'raster' in 'maia_scan_spec'.
2. Open_vsub:
 - a. Req_vsub:
 - i. Return 'seqno' as a new keyword 'seq'.
3. Scan_list:
 - a. Test old daemon:
 - i. Use 'test_stage_progress' to loop through reading
 - b. GUI:
 - i. Add "Append" button and a 'strokes' text widget display (no edit).
 - c. Pstate setup:
 - i. Add 'stroke' to 'scan_sequence'.
 - ii. Add 'seqno' to 'scan_sequence.move'.
 - iii. Add 'seqno' to 'scan_sequence.scan'.
 - iv. Simplified 'scan_sequence' in pstate to keep pause, lock_pause, raster_on, error and stroke at the top level (so they can be set using hash key-value, if moved to the KVS).
 - d. Timer events:
 - i. Correct 'scan' template:
 1. 'scan' had been changed by me to 'key' a year ago, according to the MM Confluence change history!
 2. Added 'stroke' for completed X-line strokes.
 3. Added 'duration' and 'remaining', though not used yet.
 4. NOTE: 'key' is NOT set, as it should, to the hash name.
 - ii. Update 'scan_sequence.stroke' from 'scan' PUB.
 - e. Timer loop to control phases of 'start' and 'next'
 - i. Triggered off 'skip' button timer.
 - ii. Uses phases in pointers (*pstate).phase.pstart and (*pstate).phase.pnext
 - f. scan_list_start:
 - i. If /append then copy 'scan_sequence.stroke' into selected scan entry for 'stroke'.
 - ii. Use 3 phases for move to origin, set Maia, set scan_sequence parameters
 - g. scan_list_next:
 - i. Set 'scan_sequence.raster.stroke' to 0.
 - ii. Use 3 phases for move to origin, set Maia, set scan_sequence parameters
 - h. Sequence numbers:
 - i. Although implemented above in REQs, there is no way to get them back in PUBs linked to the client that issued the REQ.
 - ii.
 - i. Version 3.2
4. GeoPIXE:
 - a. Image, Kill in all planes
 - i. Image_Process_Suppress_Region:
 1. Streamline the kill /all option for speed.
 2. Still takes a long time in 'set_image_minmax'.
5. Image_save_all_TIFF, export_images_csv:
 - a. Call 'image_flux_charge' to get charge and pixels, which accounts for non-uniform charge maps and borders better, as used widely elsewhere.

15 Aug, 2018

1. Startupp:
 - a. Added a retry on error catch to get past an X server error/ bug on chinook at BNL. See comment lines: “;@8-18”
 - b. Same for: Blog_browser, warning, startupg_last.
2. Select_evt_file:
 - a. Hack to allow blog data to extend into a second run number series.
3. NSLS_MARS_Ge device:
 - a. New device NSLS_MARS_Ge device for binary data from MARS Ge system.
 - b. Two long words per event:
 - i. First is E,T,adr in Maia event_1 format
 - ii. Second: long word time stamp
4. New GeoPIXE version 7.4u
5. New Maia-Control version 7.4u

11 July, 2018

1. Maia_launch:
 - a. Add free disk space to “Project” line, using ‘(*pm).control.status.fs_free’.

3 July, 2018

1. GeoPIXE:
 - a. Select_evt_files:
 - i. Permit the ‘finish’ file to be in the next (numerical) run directory.
 - ii. Append all files in finish dir, but use an ‘offset’ to make their seq# larger.
 - b. Version 7.4t
2. Maia_launch:
 - a. maia_launch_overdisk:
 - i. Veto disk test if Maia ‘discard’ is ON.
 - b. Version 7.4t

16 June, 2018

1. Blog_browse:
 - a. Do not assign the ‘a’ axis variables in the TAG #48 DAQ ET records with time stamps.

7 June, 2018

1. Maia device:
 - a. trim_evt_files:
 - i. If number of files exceeds the YLUT file table, then pop-up a warning to delete YLUT so that a new one will be regenerated, and then truncate the file list and continue.

31 May, 2018

1. parallel_client:
 - a. Trim username after “@” using ‘trim_user()’ to make ‘prefix’ for shared memory name usage.
 - i. This tames effects of crazy long usernames at UMelb.
 - b. Increase timeout on child startup from 30s to 60s
 - i. to cater for long IDL license service time on VMs.
2. GeoPIXE:
 - a. Trim_user():
 - i. Simple routine to trim a string after a “@”.
 - b. Raw_count_rate_image_plugin:
 - i. Plot a histogram of count-rate.
 - c. Version 7.4s

3. Maia control:
 - a. Write identity and prefix at start-up.
 - b. Version 7.4s
4. DAQ control:
 - a. Daq_launch needs to Notify Master using a different pointer 'pmaster' to avoid conflict with 'pcom' command Notify.
 - b. Version 7.4s
5. Maia device:
 - a. Get_Maia_details:
 - i. Need to look in 'metadata.scan.info' ('scan_info') for comment.

25 May, 2018

1. Maia control:
 - a. Blog Browse:
 - i. Need to call it using "blog_browser" to find SAV file.
 - b. New version 7.4r
2. DAQ control:
 - a. Blog Browse:
 - i. Need to call it using "blog_browser" to find SAV file.
 - b. Daq_scratch.def
 - i. Add 'scratch_master' = 23
 - c. Daq_launch:
 - i. Init
 1. Write "DAQ:master" to 'scratch_master' index
 2. Save local ID to 'local_id'.
 - ii. Respond to notify 'master-set' from stage_list
 1. Bracket with set lock_master
 2. Set DAQ:master in Klee, if master set.
 - iii. Timer loop to read DAQ:master from Klee
 1. Ignore if lock_master set.
 2. Set master mode if name matches local_id
 3. notify 'master' to stage_list.
 - iv. (*pstate) variables added for master mode:
 1. local_id: local ID string to be used as master ID
 2. master: flags local instance as the master
 3. prev_id: previous master, else blank
 4. lock_master: lock out timer changes while processing notify from Stage_List.
 - v. New version 7.4r
 - d. Stage_list:
 - i. Wait 3.0 s at start of 'stage_list_wait_on_raster' and 'stage_list_wait_on_move' to make sure raster/move has started.
 - ii. Check-box for "Master"
 1. Bracket with set lock_master
 2. Notify 'master-set' to daq_launch.
 - iii. Respond to notify 'master' from daq_launch.
 1. Ignore if lock_master set.
 2. To set check-box.
 - iv. (*pstate) variables added for master mode:
 1. master: flags local instance as the master
 2. lock_master: lock out notify changes while processing check-box event.
 - v. New version 3.0

14 May, 2018

1. GeoPIXE:
 - a. Image_table:

- i. Add “Centroid X” and “Centroid Y” to display mode droplist.
 - b. Image_table_eventcb:
 - i. Load_image_table_table:
 1. Add ‘Xcent’, ‘Ycent’ arrays and display in table from ‘centroid’ struct already in results.
 - c. Define:
 - i. Add ‘poly’ to ‘source’ struct definition:
 - ii. poly: { mode: 0, \$; poly mode (0=off, 1=on)
 1. gain: 21000., ; flux gain
 2. energy:17.4, ; energy of gain
 3. model: 'XOS default', ; name of transmission model
 4. diameter: 2.0, ; diameter of beam at exit (mm)
 5. focus: 14.0, ; focal distance (mm)
 6. spot: 0.03, ; focus spot size
 7. pinhole: 0.025, ; pinhole diameter
 8. distance: 100.} ; distance of pinhole
 - iii. keep old ‘source’ as ‘old_source’.
 - d. Read_source, write_source:
 - i. Read/write new source struct w/ poly
 - ii. Version -6
 - e. read_yield, write_yield:
 - i. Read older files using ‘old_source’ struct.
 - ii. Return using new ‘source’ struct.
 - iii. Version -9
 - f. Evt_start:
 - i. Comment out some constraints on ‘xoffset’, ‘yoffset’ and ‘x_sub_range’, ‘y_sub_range’ to allow setting a “Windowed” sort out of the bounds of the full sort.
 - ii. This alone still fails as the border clear vetoes events beyond normal range.
 - iii. Also extend xrange and yrange if the sub-range plus offset is beyond this normal range.
 - iv. Tag with comment “@23-5-18”
 - g. Da_evt:
 - i. N.B. The constraint of YLUT table Y values all being zero or above, means that the negative Y offset sort MUST be done with NO cluster mode.
 - h. New version 7.4r
2. Source continuum:
 - a. source_calculate:
 - i. Use XOS poly transmission and pinhole solid-angle
 - b. source_setup:
 - i. Change mono droplist to an optics droplist.
 - ii. Add (mapped) widgets for poly Gain, Energy, Pinhole (diam.)
 - iii. Read/update these in ‘source_update_pars’, ‘source_setup_pars’.
 - iv. Change OnRealize to new optics droplist.
 - v. Add events for new widgets, and change ‘mono-mode’ to ‘optics-mode’ to set both ‘(*p).mono.mode’ and ‘(*p).poly.mode’.
 - vi. Add totals to ‘source_draw’.
 - c. New yields ideally would use a “conv” of 0.001 (convert ms to s) if the model is perfect.

11 May, 2018

1. Scan_list:
 - a. Tried ‘scan_list_wait_on_scan’ to wait for raster to start or to stop, in analogy to a similar successful approach after move to origin using ‘scan_list_wait_on_move’. But it would not work. The VSUB values updated very slowly only after a minute or more. These remain commented out.

- b. Tried a 2nd VSUB 'vsub_stage2' to see if that worked, which it didn't. Could remove this vsub as well as 'comms5' object.
- c. Added a 15s wait between stop and next in skip (Notify and button).

7 May, 2018

- 1. Source_continuum:
 - a. Source_setup:
 - i. source_update_pars:
 - 1. Only set density for pure elements, if '(*p).filters[i].microns' ne 2 (NPT Gas layers) where it has been set in 'source_update_density' already.
 - ii. Add /Test to 'source_setup' launch, which will print the various Dose calculations on the terminal.
- 2. GeoPIXE:
 - a. Filter_setup:
 - i. Only set density for pure elements, if '(*p).filter[i].microns' ne 2 (NPT Gas layers) where it has been set in 'filter_update_density' already.
 - b. detector_setup:
 - i. Only set density for pure elements, if '(*p).filter[i].microns' ne 2 (NPT Gas layers) where it has been set in 'detector_update_density' already.

7 May, 2018

- 3. Scan_list:
 - a. Make Excillum '(*pm).status.power' now "-1" by default, so a lack of connection to the Excillum daemon does not force a pause.
 - b. Only =0 if connection is good, but power read-back is low.
- 4. GeoPIXE:
 - a. Open_vsub:
 - i. Enabled more detail on error messages.

24 Apr, 2018

- 3. GeoPIXE: alarm_popup:
 - a. Accepts Notify, 'cancel' to cancel the popup.
- 4. Image_table:
 - a. Popup warning before separating hot-spots to show # estimated.
 - b. Cancel will not do it.
- 5. Scan List:
 - a. Define a new variable in '(*pstate).scan_sequence.raster' called 'lock_paused'
 - i. Normally this is set in parallel to 'pause'.
 - ii. Call 'scan_list_pause' with /soft to only set a soft pause, which can be resumed automatically when beam power returns.
 - iii. This allows a manual Pause to 'stick' until Resumed.
 - b. If dx, dy don't change, calls 'alarm_popup' to report "Stage Frizen?".:
 - i. If dx,dy change enough, cancel the frozen stage popup.

24 Apr, 2018

- 1. KVS:
 - a. Added 'source' endpoints ('MM.Mel.EX.endpoints') to 'MM.Mel.SL.config'
- 2. Scan List:
 - a. Opens a VSUB to Excillum source, using endpoints in KVS.
 - i. Uses another 'comms' port 'comms4'.
 - ii. Adds parameters in state to parallel stage, etc.
 - iii. Subscribes to ['var.rt.source','alert']; 'var.change.source' will not work.
 - b. Also opens all this in 'Scan_list_retry_kvs'.
 - c. Add 2 more LEDs for 'source' and 'power'
 - d. Timer tests:

- i. 'start-button': (only if raster on)
 - 1. If power low and not paused, then pause
 - 2. If good power and paused, then resume.
- ii. 'coords-src-mode':
 - 1. Gets 'generator_emission_power' from Excillum VSUB and compares with 95% of 'generator_max_power_at_settings'.
 - 2. Power is good, if above this.
 - 3. Sets the new LEDs 8,9.
- e. Set default SRC cords to Mapper Coords using end-station index in '(*pm).endstation'.
- f. Test for sequence active, which means local control.
 - i. If not local, do not show current scan time remaining.
- g. Version 2.1

18 Apr, 2018

- 1. GeoPIXE:
 - a. Define:
 - i. Add 'bpinterlock_downtime' to 'control.status'
 - b. Temp_name()
 - i. Returns a unique temp file name.
 - c. maia_launch_interlock:
 - i. Returns the downtime as a string
 - ii. This is added to the popup by 'Maia_launch'.
 - d. Alarm_popup
 - i. Scales widget width, font size and thickness according to the number of characters in message.
 - e. Version 7.4q
- 2. Blog_browse
 - a. Uses the unique temp file-name
 - b. Deletes the temp file on exit.
- 3. maia_client_parameters_slow:
 - a. Read 'bpinterlock_downtime' → 'control.status'
- 4. Maia_launch:
 - a. maia_launch_interlock:
 - i. Check whether '(*pm).control.status.bpinterlock_downtime' occurred in the past 10 seconds.
 - ii. Only if it is not zero, and only if it is after 10s after main uptime.
 - iii. Also returns the down-time string.
 - b. Version 7.4q

12 Apr, 2018

- 1. Maia_setup:
 - a. 'hermes_elk_mode' widget ID is not defined, as it was moved to debug.
 - i. Comment out use of it in HERMES and Control sections.
 - b. New version 7.4p.
- 2. DAQ Control:
 - a. Stage_edit:
 - i. Changed the list of 'overscans' to even only, and save in 'pstate'.
 - b. Build_raster_list4:
 - i. Make variable 'overscan' equal to half of the passed overscan value as this gets halved to drive the raster, where overscans get doubled anyway.
 - ii. Set 'mixed_dac_stage' = 1 always to make the overscan always repeating the fast axis first, before advance slow axis, even if stage-stage scan.
- 3. MM_scan_list:
 - a. Scan_list:
 - i. scan_list_start:

1. Added 'scan_list_wait_on_move' to move to origin.
- ii. scan_list_wait_on_move:
 1. Add an initial wait=1.0 to allow the previous move to start before checking.
- iii. New version 2.0

9 Apr, 2018

1. Falconx device:
 - a. read_falconx_segments:
 - i. Fixed inconsistent common block with 'read_falconx_header'.
 - ii. Added 'tick' variable to end.
 - b. Still got common block extend error.
 - i. Indicates that somehow 'read_falconx_header' or 'get_header_info' method is being executed on define?
 - ii. Recompiled all device drivers and deleted un-needed sav files.
 - iii. Eventually error went away – mysterious.
2. String_packer
 - a. Unstringify:
 - i. Add string print out on errors.
3. build_raster_list4:
 - a. Try to enclose 'metadata.scan.info' string in extra double quotes.
 - b. This does not work, remove these.
 - c. Appears it is a problem with varsh, which reads correctly only at times.
 - d. Stop using 'vel_stage_save' as (*pm).stage.velocity is normal vel now.
4. Daq_client_parameters:
 - a. Need to 'socket_command_get' 'metadata.scan.info' using /string option.
 - b. Read 'scratch.datum[scratch_velocity].value' for velocities.
 - i. Save these in '(*pm).stage.velocity'.
5. Stage_list:
 - a. Stop using 'vel_stage_save' as (*pm).stage.velocity is normal vel now.
 - b. New version 2.2
6. Daq_launch:
 - a. Accept new tags in 'scan-setup' for normal velocities:
 - i. 'STAGEX_VELOCITY', 'STAGEY_VELOCITY', ...
 - b. New version 7.4q

6 Apr, 2018

1. DAQ-launch:
 - a. Stage_edit:
 - i. Copy correct list of overscans to event and update routines.
 - b. Stage_list:
 - i. stage_list_do_scan:
 1. Now calls 'stage_list_wait_on_raster' to wait for raster move to origin to complete.
 - ii. stage_list_wait_on_raster:
 1. Wait for 'status.raster.state' to become "stop".
 - iii. build_raster_list4:
 1. /set_origin mode now builds a raster list for the absolute stage and relative DAC moves to origin and runs this, which does the half-range DAC/stage offset trick.
 2. Raster charge.min and charge.max now uses float rather than integer type.
 - iv. New version 2.1
 - c. New version 7.4p

4 Apr, 2018

2. DAQ-launch:

- a. Daq_update_parameters2
 - i. Only copy (*pm).stage.velocity from background process, not the whole (*pm).stage struct, which overwrites velocity max values.
- b. Stage_edit:
 - i. Copy correct list of overscans to event and update routines.
- c. Stage_list:
 - i. New version 2.0

27 Mar, 2018

1. DAQ 36 device:
 - a. Daq_default
 - i. Add debug lines for DAQ, Maia and blob
2. DAQ control processes:
 - a. Activity:
 - i. Add debug from daq_default() and gprints to log file.
 - b. ET:
 - i. Add debug from daq_default() and gprints to log file.
 - c. Parameters:
 - i. Add debug from daq_default() and gprints to log file.
3. Maia control processes:
 - a. Parameters (norm and slow):
 - i. Add debug from maia_default() and gprints to log file.
4. GeoPIXE:
 - a. Progress
 - i. Added more space for %done write
 - b. Image_table
 - i. Add button "*" for "Star_Button" to separate highlighted pixels in clusters
 - ii. Add tracking help to all widgets.
 - iii. Add a Help widget for tracking help.
 - c. Image_table_event
 - i. OnButton_Image_Table_Star:
 1. Use the Watershed() function to build an index for each highlighted pixel cluster, and then construct regions for each.
 2. Insert these into the table and do an Update All.
 - ii. Add a tracking routine to display uvalue text in Help widget.

22 Mar, 2018

1. GeoPIXE define:
 - a. Added stage struct (w/ velocity) to both daq_struct and daq_shared structs.
2. Daq_client_parameters:
 - a. Reads 'stage.axis[].velocity' often, copies into (*pm).stage.
3. Daq_update_parameters2:
 - a. Copies (*pdat).stage → (*pm).stage
4. Stage_list:
 - a. build_raster_list4:
 - i. Saves velocity in a common (vel_stage_save) before raster start.
 - b. Stage_list stop and next:
 - i. Restore velocity from common on a stop or end of sequence.
5. Daq_launch:
 - a. Stop button also sends a Notify to Stop raster to stage_list.
6. Daq_version:
 - a. Version 7.4o
7. GeoPIXE version:
 - a. Version 7.4o
8. Maia_version:

- a. Version 7.4o

21 Mar, 2018

1. Daq_client_activity:
 - a. Added support for summary records, to see blog 'runno'.
2. Daq_version:
 - a. Version 7.4n
3. Stage_list:
 - a. Version 1.9

14 Mar, 2018

1. Maia Mapper scan_list:
 - a. 'start-button' timer action:
 - i. Increase delay veto time from 6s to 12s before checking for 'stopped' at end of run to ensure that run has started first.
 - b. scan_list_wait_on_move:
 - i. New routine to wait for Move to complete.
 - c. scan_list_next:
 - i. Use 'scan_list_wait_on_move' to wait for move to origin to complete.
 - d. Version 1.9
2. DAQ stage_list:
 - a. stage_list_start:
 - i. calls 'stage_list_do_scan' to do the scan.
 - b. stage_list_do_scan:
 - i. Call 'build_raster_list4, / set_origin' to position at origin first.
 - ii. Send maia set-up settings and metadata ('stage_list_maia_scan')
 - iii. Then build raster and start it ('build_raster_list4').
 - c. build_raster_list4:
 - i. Set origin only, if /set_origin
 1. Use 'stage.axis[axes].onpos' to test for completed move.
 - ii. Set stage origin using 'stage.axis[0-1].position.target'
 - iii. This only makes sense for stage only raster for now.
 - d. stage_list_maia_scan:
 - i. Maia set-up settings and metadata
 - ii. Set 'encoder.axis[0-1].scale' = 'pixel.dim[0-1].pitch' = pitch
 - iii. Set 'encoder.axis[0-1].position' = 'pixel.dim[0-1].origin' = origin
 - iv. Assume dwell is just 'raster.time.min' for now.
 - v. Still needs to set a crossref instead of dummy 'key'.
 - vi. Leave hysteresis at zero for now.
 - vii. Maia pixel parameters order oddities:
 1. Need to set in this order:
 - a. Encoder.axis[].scale
 - b. Pixel.dim[].pitch
 - c. Pixel.dim[].origin
 - d. ~~Wait, 0.5~~ (forgot to wait for org move)
 - e. Encoder.axis[].position
 - f. Pixel.dim[].origin (to get pixel.coord correct too)
 - e. stage_list_stop:
 - i. Endrun on Maia too.
 - f. stage_list_next:
 - i. Now calls 'stage_list_do_scan'.
 - ii. Endrun on Maia too at end.
 - g. stage_list_pause:
 - i. Photon enable on Maia too.
 - h. Notes on blog run numbers:

- i. These are known only after runs start, separately for DAQ and Maia bloggers.
 - ii. They are read periodically in ‘activity’ back ground processes and passed to foreground *pm struct by ‘...update2’ routines.
 - iii. ‘Stage_List’ reads them directly as ‘blog.runno’ from both DAQ and Maia ports, and adds these to the List table as ‘blog’ and ‘crossref’ columns.
 - iv. Note that the crossref in the metadata is not used, as this is written to blog file before both DAQ and Maia run-numbers are all known.
- i. Version 1.9
- 3. Python object close: try to better release socket open file resources.
 - a. Occasionally run out of ports on ‘scan_list_retry_kvs’.
 - i. Seems that ports are not fully closed/ deallocated after obj_destroy?
 - b. Close_vsub: explicitly delete object before IDL ‘obj_destroy’
 - i. `r = vsub.__del__()`
 - ii. Did not crash, but comment out for now ...
 - c. Close_comms: explicitly delete object before IDL ‘obj_destroy’
 - i. `r = old.__del__()`
 - ii. No, deleted this line – causes crash.
 - d. Close_kvs: explicitly delete object before IDL ‘obj_destroy’
 - i. `r = kvs.__del__()`
 - ii. No, deleted this line – causes crash.

6 Feb, 2018

- 1. Pulser_FWHM_map_spectrum_plugin:
 - a. Plugin like Fit results FWHM map, that uses X or View markers to define a pulser peak and plots a map of noise FWHM and a histogram.
 - b. Refined the algorithm to avoid counts away from peak, which upsets variance calculation. This does not work well if there is no peak found, which defaults to a large value.
- 2. Centroid:
 - a. Function to return a weighted centroid and variance.
- 3. Source_setup:
 - a. Make the pars.continuum default to 1, so it will run stand-alone
- 4. Test_excillum_tube_spectra4:
 - a. ?
- 5. Alarm_popup:
 - a. Make box longer for long text title.

2 Feb, 2018

- 1. Maia_setup:
 - a. Added a write of ELK-map to disk after completion of ELK mapping procedure.
 - b. Note that ELK is now in V; the conversion of V \rightarrow pA is not being used at the moment.
 - c. Added ELK selection to the Debug tab OAN output droplist.
 - i. Needed to keep track of two vectors of selection, one for EAN and one for ELK.
 - ii. Need to test EAN and ELK on change of tab (readback from Maia) to decide where to set droplist and section on detector mimic.
 - iii. Code in ‘maia_setup_update_debug’ to set Maia parameters.
- 2. maia_setup_write_leakage:
 - a. Write *pelk array to ‘ELK-map.csv’ in home/.geopixe dir.
- 3. maia_setup_correct_leakage:
 - a. Still skips conversion of V to pA, as the values should be >2.55 V, which they are not.
- 4. New version: 7.4m

1 Feb, 2018

- 1. Maia_setup:
 - a. Move Auto save of RT images to a separate box on Imaging tab with its own “Apply RT Save” button.

- b. New 'imaging-apply-rt' event copies local *pl copies of auto flag and save_path to *pm and updates Enable tab enable checkboxes.
 - c. Set checkbox and save_path from *pm on change of tab.
 - d. Apply HYMOD does not include save_path in stringify struct sent to Hymod as scratch.datum variable.
- 2. Maia_launch_read_da_info:
 - a. Does not look for save_path in stringify struct from scratch.datum.
- 3. Define:
 - a. Added 'status.blog_error' to 'maia_struct' struct definition.
- 4. Maia_client_parameters:
 - a. Readback 'status.blog.error' and fill (*pdat).status.blog_error.
- 5. Maia_update_parameters:
 - a. Copy 'status.blog_error' from *pdat and fill (*pm).status.blog_error.
- 6. Maia_launch:
 - a. Add to (*pstate).help string in tracking and timer event for setup.
 - b. If '(*pm).status.blog_error' is not blank, add an extra line to status window.

30 Jan, 2018

- 1. Define:
 - a. Added 'aux' to 'maia_control' struct definition.
 - b. Added 'status.fs_free' to 'maia_control' struct definition.
- 2. Maia_client_parameters:
 - a. Readback 'status.scepter.aux' and fill (*pm).control.aux.
- 3. Maia_setup:
 - a. Move OAN from Controls to Debug tab.
 - b. Add AUX text field to Debug tab.
 - c. Add a DAM Monitor select and value display to debug tab.
 - i. Add pstate fields for 'monitor_modes', 'controls_monitor_text', 'debug_monitor_mode'.
 - ii. Add Notify 'maia-display' line to set 'controls_monitor_text'.
 - iii. Add Debug change of tab to set the 'debug_monitor_mode' & droplist.

29 Jan, 2018

- 1. Socket_get_command:
 - a. Add "/L64" and "/UL64" options for Long64 and Ulong64 types.
- 2. String_packer:
 - a. Added functions "long2_64" and "ulong2_64".
- 3. Define:
 - a. Added 'status.fs_time', 'status.fs_size' to 'maia_control' struct definition.
 - b. New version: 7.4l
- 4. Maia_client_parameters:
 - a. Disk space alert:
 - i. Monitor 'status.blog.fs.free' for disk space and 'status.blog.fs.time' for time remaining in hours (at current rate).
 - ii. Perhaps focus on 'fs.time' for an alert (e.g. when time is less than 100).
 - iii. But ignore it all if 'fs.size' is zero, which means that these data are not available.
 - b. Added read of 'status.blog.fs.size' and 'status.blog.fs.time'.
 - c. Copy these into (*pm).control.status struct.
- 5. Maia_launch:
 - a. Added 'maia_launch_overdisk' test to timer loop on setup button.
 - b. New version: 7.4l

19 Jan, 2018

- 1. DAQ_defaults:
 - a. Add "maia" parameters, as in Maia device/defaults.

2. DAQ_launch:
 - a. Also open a slave Maia port *psm, if IP non-blank in conf file.
 - i. 'psm' saved as 'pmsocket' in state.
 - b. Now use LED[4] for Slave Maia socket open.
 - c. New version: 7.4k

18 Jan, 2018

1. Maia detector identity behaviour changed.
 - a. Now assumes that identity in DAM will set the log file name by default.
 - b. File will be opened in same dir as .Maia.conf file.
2. Maia device:
 - a. Maia_defaults:
 - i. 'log_file' remains blank unless explicitly overridden in .Maia.conf file
3. Maia_setup:
 - a. New 'show parameters' check box for Scepter 'Thresh – Trim', which is the "overall threshold".
 - b. Added 15 to 'check_ids' array for this checkbox ID.
 - c. Add 15 to 'check_ids_scepter_channel' vector.
4. Maia_launch:
 - a. If 'log-file' is blank (from 'maia_defaults') then set file-name based on conf file path and name built from identity.
 - b. If 'log-file' is NOT blank, it is assumed that an explicit one has been given in conf file as an override.
 - i. Pop-up a Warning, if this happens.
 - c. New version: 7.4k

11 Dec, 2017

1. Scan_edit:
 - a. Event: 'set-origin-button'
 - i. Add a "Set" button next to origin to enter it in "Ruler" coordinates.
 1. $X = X^* + 4.07$
 2. $Y = Y^* - 6.68$ for X^*, Y^* in ruler cords
 - b. scan_edit_update:
 - i. Add silent keyword to pass to 'scan_list_check_bounds'.
2. Scan_list:
 - a. scan_list_check_bounds:
 - i. Test all limits before return bad error
 - b. Add "Skip" button:
 - i. Does a 'stop' (/no_complete) and then 'next'.
 - c. Add remaining time for current, as well as all.
 - i. Set these in 'update_scan_list_time'.

21 Nov, 2017

1. test_excillum_tube_spectra4:
 - a. Now use empirical XOS polycapillary transmission.
2. xos_transmission:
 - a. empirical XOS polycapillary transmission function (E).

17 Nov, 2017

1. Linearize_cuts_energies_spectrum_plugin:
 - a. Now assumes an initially calibrated spectrum, which makes sense for this plugin, and just fits the channel error between a peaks centroid and it's expected energy.
 - b. The CUTs need to be named "name energy", with the 'energy' separated from the 'name' by white space.

15 Nov, 2017

1. image:
 - a. Add 'Kill-Region-all' to kill a selected region in all element planes.
2. Image_eventcb:
 - a. Image_Process_Suppress_Region:
 - i. Add /all switch to loop as well as for XANES.

6 Nov, 2017

1. Define:
 - a. 'maia_shared1':
 - i. Add 'identity.dam' byte array.
2. maia_client_parameters:
 - a. Read 'config.dam.identity' from Kandinski, copy to shared pm identity.dam.
3. Maia_update_parameters:
 - a. Copy shared memory pm identity.dam to *pm.
 - b. Pop-up warning of identity changes.
4. Maia_launch:
 - a. New version: 7.4j

25 Oct, 2017

1. Date_from_utc:
 - a. Extend 'sec' for display fractional seconds.
 - b. Remember to pass a 'double' UTC argument to this.
2. Maia_launch:
 - a. Maia_launch_interlock:
 - i. Corrected test on bpinterlock uptime to look for uptime in last 10 seconds from current time.
 - ii. However, it now ignores a situation when current time minus uptime is negative, which suggests that the Hymod has a bad time set.

17 Oct, 2017

1. EVT:
 - a. Evt_set_evt_file:
 - i. Do not use /set for 'build_output_path' here.
2. FalconX device:
 - a. Build_falconx_ylut:
 - i. If no Y found in first buffer (0.2 MB), then rewind and try a 6 MB buffer.
 - ii. Only look for 'ylut_accept' tags, which are defined in 'falconx_listmode.def'.
3. Fxbrowse:
 - a. Added more buffer sizes to droplist.

16 Oct, 2017

1. Sub-region select in Sort EVT ("Get" button) or in Scan Edit, needs:
 - a. Setting the KVS 'newscan' to the selected Box (mm).
 - b. Setting 'region_window' in common to the Box (pixels).
2. Image_routines:
 - a. Set_kvs_box:
 - i. Writes box (mm) to KVS.
3. OnButton_Image:
 - a. Use 'set_kvs_box' to write a Box to KVS.
 - b. Also write Box pixels to 'region_window'.
4. OnNotify_Image: 'image-region-select'
 - a. Use 'set_kvs_box' to write a Box to KVS.
 - b. Also write Box pixels to 'region_window'.

5. New version: 7.4j

11 Oct, 2017

4. Maia device:
 - a. Read_maia:
 - i. Fixed test on subscripts in EV for 'b'.

22 Sep, 2017

1. Maia_setup: apply hymod
 - a. Lost the line that defines 'ps'. Fixed.

19 Sep, 2017

1. qsample:
 - a. Fixed the calculation on 'ql' and 'qh', in final q[] selection, to cast to Long to keep them bound in 0 to (nq-1) range.
 - b. Also make sure that nq=0 is returned on error with q=0.

12 Sep, 2017

1. Spectrum_load_prep:
 - a. Changed action with pileup and throttle:
 - i. Issue if pileup is present in header, and exists on disk, it gets used without a file prompt. This is good, unless the pileup specified is not correct. Better to prompt for it, passing current name.
 - ii. Now will always prompt for pileup, throttle, providing current file-name, unless is was explicitly off (on=0).
 - iii. If "cancel" this passes back a blank file-name now.
2. Maia device:
 - a. Get_maia_details:
 - i. If we find a valid pileup or throttle field in scratch, then set found=1.
 - ii. Only set on=0 if either on=0 is set in Maia field or filename is blank.

11 Sep, 2017

1. FalconX device:
 - a. get_falconx_info:
 - i. Fixed bug in 'sample_serial'
 - b. update_header_info:
 - i. Fixed bug in comment.
2. Scan_list:
 - a. Added LEDs for manual, preset moves which show:
 - i. Green = moving, Red = locked, off = stopped.
 - ii. "locked" indicates that stage is locked by the Safety system.

7 Sep, 2017

1. Maia device:
 - a. maia_et2_events2:
 - i. Extended 'maia_et2_events' to include Z axis (only for ET4).
 - b. blog_client_et2_spectra:
 - i. Use 'maia_et2_events2'
 - ii. Z ignored until needed.
 - c. Blog_client_da2:
 - i. Get axis redirection from (*ppar)[6:8]
 - ii. Set x,y,z based on redirection
2. Define:
 - a. Added 'axes' vector to (*pm).DA 'Maia_struct'

3. Maia_control:
 - a. Maia_setup:
 - i. Render axes options using render_options method:
 1. For now use unique switch /axes_only to only render the axes redirection controls. This is unique to Maia device.
 - ii. maia_setup_apply_axes:
 1. Get_options, set (*pm).DA.axes
 - b. maia_update_da2:
 - i. Set (*ppars)[6:8] = (*pm).DA.axes
4. FalconX device:
 - a. get_falconx_info:
 - i. Added all 4 FC'n and 6 dim'n axes.
 - ii. Not really using 'units' as yet.
 - b. Extended to 6 axes
 - i. e.g. in 'update_header_info'.
 - c. Need FCn to be selected like axes too, within device. However ...
 - i. Only would work if FCn selector widgets were within device too.
 - ii. This would require major structural changes!
 - iii. In the meantime, the default sensitivity will always be set by FC0 only.
 - d. Falconx_version:
 - i. Also digs out digital sampling rate from "afe.sampleRate" to be used to determine the clock 'tick' used for converting sample counts to dwell time (ms).
 - e. IDL Directory organization for GeoPIXE:
 - i. Moved all files in geopixe/device/falconx into the falconx device dir.
 - ii. For this to work, any compiles need to have the **falconx device project open** (such as compiling FX_browse) so that it can find files like 'falconx_listmode.def'.
 - f. Get_falconx_details:
 - i. Now simply adds to the 'info' struct and passes this on.
 - g. update_header_info method:
 - i. Now works directly from 'info' struct, as stored in self.
5. File_requester:
 - a. Added "numeric part" checkbox for sorting by the final numeric part of a file-name.
6. Find_file2:
 - a. Added options to sort by numeric_part, final part of file-name
7. IDL sav file launch:
 - a. Added scripts to launch idl_query, maia_update, maia_control, geopixe, etc.
 - b. Templates added to SVN.

3 Sep, 2017

1. FalconX device:
 - a. Build_falconx_ylut:
 - i. If XY not found in a file, replace with Y for next file.
 - ii. Work forward, propagate Y if no XY found.
 - b. Get_falcon_ylut, write_falcon_ylut:
 - i. Look for file name, stripping off "_n"
 - c. Flux_scan:
 - i. Simple use header values – assume 'get_header_info' previously.
 - d. Read_falconx_header:
 - i. If version zero, try a "_0" segment file instead.
 - ii. Get info file, using 'Get_falconx_info'.
 - e. Get_falconx_info:
 - i. Read .info file and return full set of metadata in 'info' struct.
 - f. Get_falconx_details:
 - i. Add extra metadata to extended struct paralleling Maia device.
 - g. Read_falconx_segments:

- i. If version zero, try a “_0” segment file instead.
- h. Get_header_info:
 - i. Always read “_0” segment file, if found.
- i. Update_header_info:
 - i. Add extra header.metadata paralleling Maia device.
- j. Flux_scan:
 - i. Done simply, assume header already read in.
- k. Init:
 - i. Enable use of YLUT
 - ii. Use ‘.silist’ extension
- 2. Maia device:
 - a. Read_maia_32_header:
 - i. Uses now the ‘free_record’ to free records
- 3. Spec_evt:
 - a. Correct error in dead-fraction (mode=1):
 - i. $\text{dead_fraction}[q1] = \text{dead_fraction}[q1] / (1 + \text{dead_fraction}[q1])$
 - ii. See notes “Dead Time Correction.docx”.
 - b. Changed maximum DT to 0.95 to match others.
- 4. evt_start_image_increment2:
 - a. Ignore xmin, etc, if negative to avoid stripe with no XY and default mins set to 100000 in da_evt, which appears negative to signed shorts.
- 5. Evt:
 - a. Forget?
- 6. New version: 7.4i

28 Aug, 2017

- 1. Image_tiled:
 - a. Events:
 - i. Now reads both image and Z map for new Frame #.
 - b. Postcreate_image_tiled_base:
 - i. Now has *pz for Z map image struct
 - c. Image_tiled_routines:
 - i. image_tiled_read_image:
 - 1. Also able to read ‘FLOAT’ format images for Z maps.
 - 2. Use /Zmap switch for Z map data to change key name.
 - 3. Accepts ‘BYTE’ format in place of/ as well as ‘BINARY’
 - d. onbutton_image_tiled:
 - i. On ‘UP’ calculates new data to pass to KVS:
 - 1. Box: Z origin and Z surface bilinear coefficients to ‘position’.
 - 2. S. Pixel: Z position from Z map to ‘newscan’.
 - 3. Take Care: the routines ‘xy_to_microns’ and ‘microns_to_xy’ (in ‘image_routines’) do not know about origin XY offset, so need to offset manually.
 - e. Image_routines:
 - i. ‘microns_to_xy’, ‘xy_to_microns’ and ‘xy_to_pixel’:
 - 1. Add /Z switch to use (*pstate).pz instead of (*pstate).p
 - 2. Use this for Z maps.
- 2. Scan_list:
 - a. Add ‘zsurface’ 4xfloat coefficients to the ‘finder.newscan’, ‘geopixel.newscan’, ‘geopixe2.newscan’ struct in *pm.
 - i. The GeoPIXE ones will not be used yet, but the finder one will receive Z surface parameters.
 - ii. The GeoPIXE ones may be used later if each scan inherits a Z map?
 - b. Timer (‘coords-src-mode’):
 - i. Add ‘zsurface’ 4xfloat coefficients to the ‘templates’.

- ii. Set 'finder.newscan', 'geopixel.newscan', 'geopixe2.newscan' structs from KVS newscan values (from 'onbutton_image_tiled' or 'onbutton_image').
- c. Translate:
 - i. Transform XYZ using Ref ,marks.
 - ii. Approx. for Zsurface bilinear, ignores rotation, scaling.
- 3. Scan_edit:
 - a. Needs to use (*pm).current.newscan.zsurface to add Z surface parameters to the current raster *p.
 - b. Ref Mapper uses current stage position for Z.
- 4. util:
 - a. List_to_normal:
 - i. Converts simple List (or list of list) to arrays for simple types
 - b. Hash_to_struct:
 - i. Extended to detect simple Lists (for arrays) as well.
 - c. SurfaceZ:
 - i. New function to calculate bilinear surface Z from coefficients for a given XY.
- 5. Maia_Control:
 - a. Blog_client_da2:
 - i. Set new buffered mode (setblockqueuedepth, tag 60) = 1000.
 - b. Blog_client_group_spectra:
 - i. Set new buffered mode (setblockqueuedepth, tag 60) = 12.

23 Aug, 2017

- 1. Maia device:
 - a. Maia_defaults:
 - i. Add maia.debug and blog.debug to .Maia.conf files.
- 2. Maia control:
 - a. Blog_client_activity:
 - i. Pick up blog.debug from maia_defaults for conf file.
 - ii. If debug, then open log file and gprint things to it.
 - b. Blog_client_da2:
 - i. Same as activity, plus ...
 - ii. Log DA record X,Y,flux1,dwell
 - c. Blog_client_et2, blog_client_groups, blog_client_epics:
 - i. Same as activity.

7 Aug, 2017

- 1. GeoPIXE:
 - a. Spectrum_load_prep:
 - i. Need to return an updated 'sensitivity' if it changes, so it can be passed to 'spectrum_load_do' from 'spectrum_load' in 'spectrum_display_eventcb'.
 - b. 'spectrum_load' in 'spectrum_display_eventcb':
 - i. Pass 'sensitivity' variable between prep and do load routines.

26 June, 2017

- 1. Python:
 - a. Windows 10 has problems with python's ZMQ librarys pyzmq. Needed to:
 - i. conda update conda
 - ii. conda update anaconda
 - iii. conda install pymzq
 - iv. this updated python27 to 2.7.13, pyzmq to 16.0.2
 - v. See notes "New laptop firewall blockages.docx" in New Laptop dir.
- 2. Maia_control:
 - a. Add the loss alarm popup, for discard_rate > 1000 c/s.
 - b. Maia_defaults:

- i. Add “maia project_select 1” in .Maia.conf to enable “project” changes.
 - ii. Add these to Maia Mapper .maia.conf files (mr-02, mf-03, mr-04, pc).
- 3. Scan_list:
 - a. scan_list_maia_scan:
 - i. Now writes hysteresis to Kandinski for each scan started.
- 4. Version 7.4h

1 June, 2017

- 1. GeoPIXE:
 - a. Conc Offset Image Plugin:
 - i. New plugin to simply offset a map by a fixedconc (ppm).
- 2. Mm_scan_list:
 - a. Change all references to dwell to “ms”:
 - i. Including read-write of scan details to the KVS.
 - ii. Except import/export to files, where dwell remains in “s”.
 - b. Changes to Stage commands:
 - i. “scan” replaces “raster” command.
 - ii. PUB returns for scan now:
 - 1. Changed “SCAN” to “KEY”.
 - 2. Added: “DURATION” and “REMAINING” times.
 - 3. See MM Confluence.
 - c. New version 1.7.

16 May, 2017

- 1. GeoPIXE:
 - a. Linearize CUTs named Energies plugin:
 - i. Can accept now “label energy” or “energy” names for each CUT.
 - b. Linear in View plugin:
 - i. Can select polynomial order now (5, 7, 9th order).
 - ii. 5th is still recommended.
 - c. Map_spec:
 - i. Fixed bug that didn’t handle lack of cal properly.
 - d. Layer_setup:
 - i. Illegal “group” parameter to ‘file_requester’.

3 May, 2017

- 1. GeoPIXE:
 - a. image_reset_display_range:
 - i. set_image_minmax:
 - 1. Change done recently does not handle old images without bounds data well.
 - 2. Now treat cases with apparently good bounds separately to null bounds cases.
 - b. Fonts throughout GeoPIXE
 - i. Need to stop using the “widget_control, default_font=’” function and rely solely on default OS fonts.
 - ii. This existed in several places (now marked by “;@2” at the start of the line to comment it out:
 - 1. daq_rates (daq_launch, daq_setup were already commented).
 - 2. detector_setup, evt, image, layer_setup, progress, Corr, filter_setup, flux_select, imagergb, layer_plot, pca_cluster, spectrum_display, time_amp
 - 3. maia_rates (maia_launch, maia_setup already commented).
 - 4. image_tiled,
 - 5. scanning
 - 6. source_setup.
 - iii. Also commented out new test code that set the font manually for table widgets is:
 - 1. Fit_setup, image_table, spectrum_select.

- iv. Blog_browse (also browsers for FalconX, MDAQ2)
 - 1. Still uses a manual font for the lists.
 - v. Manual fonts still in use for:
 - 1. Identify2, periodic table
2. Version: 7.4g

18 Apr, 2017

- 1. GeoPIXE:
 - a. Da_evt_stack:
 - i. Sometimes Z pitch does not match the pitch in the 'translate_energy' vector. Need a different approach that looks at actual Z pitch used.
 - ii. Approach assumes for now that axis is DCM angle and we use a lookup table to get beam energy. Must cater for case when pitch in angle does not necessarily match the lookup table file. Hence, we calculate the angle from the pixel index using the actual pitch found, and then look for this angle in the 'angle' table and adopt the matching energy. This gives us the 'translate_energy2' table, which is all energies referenced in the raw data Z pixels.
- 2. Version: 7.4f

8 Apr, 2017

- 1. GeoPIXE:
 - a. Get_maia_details (device/maia):
 - i. For short blog files the 2 MB read went past end of file and some (*pr[i]).b payloads came up short and index access to 'b' throws errors.
 - ii. Now test for remaining bytes enough for needs of record decode, else skip to monitor processing.
- 2. Version: 7.4e

5 Apr, 2017

- 1. GeoPIXE Fortran:
 - a. maia_384_events6:
 - i. New flag 'xy_on' that is only set when the scan raster X enters either the left or right border (set by the 'clear' width) and Y within 0-2.
 - 1. Only when 'xy_on' is set do we test for in left/right border
 - a. If fresh, increment Y
 - b. and set the 'ldone' or 'rdone' flags.
 - 2. Then set Y to ty.
 - ii. If outside borders (+1), then clear the 'ldone' and 'rdone' flags.

23 Mar, 2017

- 1. Geopixe_update:
 - a. Default to pftp.csiro.au and /update/7.3
- 2. ftp_defaults:
 - a. Default to pftp.csiro.au and /update/7.3
- 3. Version 3.8

21 Mar, 2017

- 1. Fortran:
 - a. FalconX_events3:
 - i. Length parameters: 'position_length', 'spatial1_length' must change with version.
 - b. Lib version 32
- 2. GeoPIXE version 7.4d

14 Mar, 2017

1. DAQ 36 device:
 - a. Update_header_info:
 - i. Always return scan XY origin from physical stage ZUV origin.
 - ii. Uses the full "scan.info" struct in metadata.
2. DAQ Launch:
 - a. Logos:
 - i. New DAQ 36 logo for main window and Rates.
 - b. Resize event:
 - i. Adjust length of Help and Control Stats frames.
 - ii. Adjust ysize of buttons (27) to match list, as in Maia_launch.
 - c. Send 'endrun' after scan goes to 'STOP' mode.
 - i. This test in daq_launch timer (setup) does not appear to be executed.
 - ii. This is still a puzzle.
 - iii. Added sending the 'scan-command' 'stop' notify on 'raster_status' error.
 - d. Version 7.4c
3. Daq_client_parameters:
 - a. *pdat/ *pm scan struct
 - i. Includes extent, pitch, origin for all 6 axes in pdat.
 - ii. Includes scan.info struct string as byte array.
4. Daq_update_parameters:
 - a. Move *pdat scan struct parameters to (*pm).scan, for selected XY axes from device.
 - b. This includes scan.info.
 - c. In future can dig out strict scan origin from (*pm).scan.info → origin.
5. Stage_list:
 - a. Build_raster_list4:
 - i. Already writes a stringify version of whole scan struct to metadata 'scan_info'.
 - ii. Use this for physical stage origin because the Klee 'pixel.dim[].origin' may be strange to pixelate DAC axes.
 - b. New "Grab" button:
 - i. Appends a new scan to list based on current 'scan.info'.
 - ii. Uses the (*pm).scan.info stringify struct (read back from Klee)
6. Stage_edit:
 - a. Z origin:
 - i. Set extent (left/right, ..., centre) now grab Z origin from stageZ.
 - ii. New "Grab Z" button to just set Z origin from stageZ.
 - b. Overscans:
 - i. Added new 2, 5, etc.
 - ii. [1, 2, 3, 5, 10, 20, 30, 50, 100, 200, 300, 500, 1000]
7. GeoPIXE:
 - a. Device/DAQ: Get_daq_details:
 - i. If 'scan_info' metadata value is not blank, then set stage (ZUV) origin.
 - ii. Note: XY are DAC XY, stage XYZ are ZUV.

9 Mar, 2017

1. Image_routines: set_image_minmax:
 - a. Need to cater for tests against 'bounds' to determine suitable 'border' ignore ranges.
 - b. Corrected error where 'offset' needs to be divided by 'compress' to effectively get 'offset3' as added to min/max values in 'da_evt'.
 - c. Borders for X and Y estimated separately.
2. ftp_connect: ftp_get_file:
 - a. If subsequent parts of file have zero 'bufsize', then skip to 'fin'.
3. Geopixe_update:
 - a. Version 3.7.
4. Version 7.4c

2 Mar, 2017

1. Scan_List:
 - a. Added Z axis translation based solely on ref mark #1 in frame finder/ mapper.
2. Open_kvs:
 - a. Retry respects the ztap setting from (*pstate).ztap.
3. Socket commands get, set:
 - a. All retry pop-ups now have “cancel” option.

28 Feb, 2017

1. Scan_Edit:
 - a. Add IGSN text widget. <return> does search on Name, but uses IGSN text.
 - b. Needs a Commit on Scan_List before IGSN is written to sample in KVS.
 - c. “?” on Sample Name lists all samples; <return> on Name filters search.
2. Define:
 - a. Added “IGSN” to ‘maia_sample_spec’.

21 Feb, 2017

1. Maia Launch:
 - a. Uses ‘default.maia.timeout’ now for background process timeouts.
2. Maia Device:
 - a. Maia_defaults:
 - i. Added “maia timeout”, which defaults to 30 seconds.
3. Open_comms:
 - a. Set the ‘basic_config’ level to DEBUG here in /ztap mode.
 - b. Do this only once, for first Comms object.
 - c. Throws an error if this is done before comms object created!
4. Open_kvs:
 - a. When /ztap selected, it now uses direct python calls, and not the “test” ones.
 - b. Also, explicitly writes debug to the mr-04-mel ‘server’.
 - c. Fixed common error in close_kvs.
5. Open_vsub:
 - a. When /ztap selected, it now uses direct python calls, and not the “test” ones.
 - b. Also, explicitly writes debug to the mr-04-mel ‘server’.
6. Image_tiled (Camera_view):
 - a. Uses KVS endpoint from geopix_defaults().
 - b. Cater for PNG, JPEG, etc. encoded data in the KVS image ‘data’ Base64 encoded string field.
 - c. Added ‘Image.Format’ and ‘Coords.Span’ structs as per Confluence.
 - d. **Needed to resolve ‘idl_base64()’ in build for some reason?**
7. Scan_list:
 - a. Uses KVS endpoint from geopix_defaults().
 - b. If ztap option, use same comms object for all kvs, vsubs.
 - c. Now explicitly select subscribe tags for Safety daemon.
 - d. Take great care with ‘scan_list_retry_kvs’ to NOT make a replacement comms object or do the DEBUG and ztap set-up a second time.
8. Scan_edit:
 - a. Sample search revised: “?” lists all, while text <return> searches for first few characters matching.
 - b. Fixed error: “coord” list was missing.
9. Onbutton_image_tiled:
 - a. Add placeholders only for Z coords from Z map.

17 Feb, 2017

1. OM DAQ device: Header read:

- a. Added support for header-versions 12, 13.
 - b. Problem has emerged in the live-time values returned. Needs to be fixed.
2. Image_table_eventcb:
 - a. Fixed error in 'detector_select' switch on 'spec_xstep_evt' and 'spec_evt' calls.
3. Clone_object:
 - a. Needed tests to cope with Null objects.
4. Obj_null:
 - a. New function to test for a null Object.
5. Image_table_eventcb: OnCellSelect_Image_Table:
 - a. Does a simple plot (wset 4) for a selected row of XANES region table.
 - b. Uses Catch on a wset to test for window 4 open already, else open it.
 - c. Tests for XANES data simple by looking at first 2 'el's to see if they are float.
6. Version 7.4b

24 Jan, 2017

1. PNC-CAT HDF device:
 - a. 'check_plist_pnc_cat_hdf': fixed returning unique list of PVs based solely on headings found.
2. Maia_launch:
 - a. Increase timeout for running background processes to 5 minutes to cater for slow network at AS.

18 Jan, 2017

1. GeoPIXE:
 - a. Plot_images:
 - i. Do not use special Ftemp file for WMF; treat the same as CGM.
 - b. Set_device:
 - i. For WMF files, move 'file=file' option to the first use of the 'device' command to set the new file name correctly.

13 Dec, 2016

1. GeoPIXE:
 - a. Da_evt_stack:
 - i. If 'collapse_energy' has been set, then collapse planes onto set used for DA matrix stack using the 'e_lookup' table recreated from pz and eDA.
 1. In Cluster mode, delay this for 'evt_start_image_increment2'.
 2. Else, veto energy planes that appear blank, and trim pz_coord list as well.
 3. Report xsize as this shortened size (keep original as it was). This seems to work with "From XAN" in Sort EVT OK for reload.
 4. Fix 'stack_type' = 2 only if /collapse_energy and /cluster.
 - b. 'Evt_start_image_increment2':
 - i. If 'collapse_energy' has been set (and hence stack_type=2), then collapse planes onto set used for DA matrix stack using the 'e_lookup' table recreated from pz and eDA.
 - ii. Else: For a XANES stack, look for any blank energy planes and remove them. i.e. Compress image, error and flux.
 - iii. Also compress the matching pz_coords list, and set a new zsize.
 - iv. Use file_requester(/skip_if_exists) to locate DA matrix fgile for XANES energies, if not found.
 - v. Accept the path=path argument, for file-requester path.
 - vi. For /flatten entry, do stack_type=2 code first BEFORE flatten to flux.
 - c. Evt_start:
 - i. Add 'collapse-energy' flag for 'da_evt_stack' call.
 - ii. Add path=path to all calls to 'evt_start_increment_image2'.
 - d. Image_event:
 - i. Add path=path to calls to 'retry_image_increment'.

- e. Evt:
 - i. Add a “Collapse E” check-box to DA tab in Stack mode, to enable collapsing the Z pixel angle/energy planes onto the fixed E list in DA matrix stack.
 - ii. Fixed mapping of “Get” on scan window for sub-region.
 - iii. Fixed mapping of “New” button in MPDA mode.
 - iv. Read text strings XANES El and file before “Start”.
 - v. Fixed window open: it was not setting 3D stack mode as used previously.
- f. Define:
 - i. Stack_type = 2 defined for use as temporary XANES stack flag, as a proxy for ‘collapse_energy’, which indicates that the stack has not been collapsed onto energy yet.
- g. image_flux_flatten_stack:
 - i. Act on stack_type=2 the same as 0 for XANES.
- h. Image_details:
 - i. Add XANES ‘energies_file’ to history window display.
- i. Spectrum_load_prep:
 - i. Always prompt for the pileup file, to allow it to be changed.
- 2. Blog_browse:
 - a. Set unity charsize, charthick, thick for plots.
- 3. Version 7.4a (revision 62).

23 Nov, 2016

- 1. Maia device:
 - a. Use new accumulation of Lost and Raw counts in accumulate routines and then use get_dead_weight_mode()=1 in ‘da_evt’ rather than dwell norm.
 - b. Maia_accumulate_dtfx2, 3D:
 - i. Accumulate ‘dtsum’ and ‘raw’ in each pixel and then calculate dead-fraction ‘df’ and hence ‘lost’ counts for last pixel, when we step into next pixel.
 - ii. Must allow for event records with no ET events.
 - c. maia_accumulate_dtfx2_3D, maia_accumulate_dtfx2_3D:
 - i. New wrapper routines for new ‘lost’ and ‘raw’ accumulators.
 - d. get_dead_weight:
 - i. New method to return ‘raw’ weights.
 - e. get_dead_weight_mode:
 - i. New method to return weight mode=1 for lost/raw dead-fraction measure.
 - f. Get_maia_details:
 - i. Detect energy of 100.00 keV as a “magic number” and call ‘run_energy_lookup()’ to select a run-energy lookup table file to return an energy for a run number listed. This selection is saved in a file in .geopixe called “run-energy-lookup.conf”. This was done to help Petra files which have fixed 100 keV energy but contain XANES data.
- 2. Time_amplitude:
 - a. Initialized progress ‘p’ struct incorrectly (probably copied wrong template when adding the extra ‘progress_file’ mode recently).
 - b. Infinite re-draw problem (again)!
 - i. The redraw triggers a re-size event, which does a re-draw, ad infinitum ...
 - ii. Logged with ESRI as ticket “GBT1624928 - Unwanted 'resize' events after a redraw”.
 - iii. The resize event happens after the oplot of data in ‘draw_time_amp’. Now in IDL 8.5, this triggers a resize that gets executed immediately, even before the draw routine exits! But the resize calls the re-draw!
 - iv. This is why we used the ‘skip_resize’ flag to suppress this re-draw. But sometimes this mechanism does not work.
- 3. Maia_launch:
 - a. Resize:
 - i. Resize main window in X only.
 - b. Display of pixel address:

- i. Now shows as “axes:” and the 0,1,2 pixel coords.
- c. State_button:
 - i. Added a new attribute ‘highlight’ which should draw a white spot in the centre of the button (in ‘state_button_realize’).
 - ii. State_button_realize:
 - 1. Draw (using polyfill) an additional 3x3-pixel spot (white with black central spot) in the centre to signify the “highlighted” state.
 - iii. State_button_set:
 - 1. Add new attribute “highlight” to set and state.
 - 2. This “highlighted” attribute is distinct from the ‘select’ attribute, which is the colour index.
- d. Detector_mimic:
 - i. Acts on ‘select-highlight’ notify and sets new attributes in state buttons for ‘highlight’.
- e. Onbutton_spectrum:
 - i. Notifies of highlight spectrum select via ‘select-highlight’ notify.
- 4. GeoPIXE:
 - a. Multiplicity_scale:
 - i. New switch ‘select’ to select rG sum for a single element, as used for XANES stack (called from da_evt_stack).
 - b. Da_evt_stack:
 - i. Call multiplicity_scale for just the XANES element ‘i_select’.
 - ii. Do not change output file-name anymore.
 - iii. This is not compatible with Cluster rebuild, so is disabled for now:
 - 1. Veto energy planes that appear blank, and trim pz_coord list as well.
 - 2. Report xsize as this shortened size (keep original as it was). This seems to work with “From XAN” in Sort EVT OK for reload.
 - c. Write_geopixe_image:
 - i. Do not set IC.conversion in PIXE mode; leave at 1.0
 - d. Retry_image_increment:
 - i. Fix error catching when in /debug mode.
 - ii. Allow it to do both DAI and XAN files.
 - iii. Do ‘set_image_minmax’ AFTER flatten step.
 - iv. Form output file by stripping off any number extension.
 - a. Evt:
 - i. Add a “Get” button to Scan tab to retrieve Box offset/range for windowed sort.
 - 1. Pass this via a new common block from ‘onbutton_image’ for Box.
 - e. Evt_start:
 - i. Do ‘set_image_minmax’ AFTER flatten step for all occurrences of ‘evt_start_image_increment2’.
 - f. Maia device:
 - i. Use signed 32-bit X,Y,Z coordinates internally, and then offset and return as uint to da_evt. This allows up to 32k pixels, and using offset to go beyond this indefinitely.
 - 1. Maia device:
 - a. Changed to x0,y0,z0 as Long in IDL device common and passed to Fortran.
 - b. Changed to maia_0, maia_1, maia_2 as Long in device common and passed to Fortran.
 - 2. Maia_384_events6 Fortran (and IDL wrapper):
 - a. x0,y0,z0, x1,y1,z1 all as Integer*4.
 - ii. Da_accumulate (Fortran):
 - 1. New option ‘compress’, which forces compressing energy axis z1 down onto z2.

14 Nov, 2016

1. MDAQ2 device:
 - a. Read_mdaq2:
 - i. Mod to parallel additions to falconx:
 1. Skip whole long.
 2. Add to 'cur'
 3. 'n_actual' needs /4 for long count.
 4. Care with record length for split record.
2. Spec_evt:
 - a. Make spectrum size 10 bigger than max channel seen to start with.
3. Base device:
 - a. Add the new method 'get_dead_weight'.
 - i. Returns error by default.
 - b. Add the new method 'get_dead_weight_mode'.
 - i. Returns 0 by default.
4. MDAQ2 device:
 - a. MDAQ2_events (fortran):
 - i. Do NOT store 'de' in t(n). This is done in fx(4,n) now.
 - ii. Busy3 record found:
 1. Estimate "lost" counts in pixel as fx(4,n) using busy time and previous dwell (assumes dwell processed before busy).
 2. Return 'raw' count in fx(5,n) (get_dead_weight_mode=1)
 - iii. If not Busy3:
 1. Busy, Busy2 do NOT set weight fx(5,n).
 - b. MDAQ2_accumulate_dtfx, 3D
 - i. Use test on 'veto' for spectrum events, not pseudo.
 - ii. Accumulate lost count in pixel or detector from fx(4,n).
 - iii. Accumulate raw count in fx(5,n).
 - c. Add 'get_dead_weight' method override:
 - i. Returns the 'mdaq2_weight' array of pixel counts.
 - ii. Sets error (to veto this weight) if total zero.
 - d. Add 'get_dead_weight_mode' mode method:
 - i. Returns the counts weight mode (0=icr, 1=raw weights).
 - e. MDAQ2_browse:
 - i. Add Busy3 records.
5. Cut_evt, da_evt, da_evt_stack, da_evt_tomo:
 - a. Change high DT, PU cut-off from 0.5 to 0.95.
 - b. Retrieve 'get_dead_weight()' to normalize the returned 'dead_fraction' by raw counts, if it is counts weighted. Then do not norm to dwell.
 - c. Retrieve 'get_dead_weight_mode()' to set approach.
 - d. If mode=1, then $df = df/(1+df)$
 - e. See "Dead Time Correction.docx".
6. Spec_evt, :
 - a. Use new counts weights and 'get_dead_weight_mode()' approach.
7. Strip_clip:
 - a. Fixed spelling 'slient' → 'silent'.
8. Falconx device:
 - a. falconx_events3 (Fortran):
 - i. Decode 'icr' and 'raw' count for pixel and derive lost counts, returned in fx(4,n).
 - ii. Return 'raw' count in fx(5,n) (get_dead_weight_mode=1)
 - b. falconx_accumulate_dtfx, 3D:
 - i. Accumulate lost count in pixel or detector from fx(4,n).
 - ii. Accumulate raw count in fx(5,n).
 - c. Add 'get_dead_weight' method override:
 - i. Returns the 'falconx_weight' array of pixel counts.

- ii. Sets error (to veto this weight) if total zero.
- d. Add 'get_dead_weight_mode' mode method:
 - i. Returns the counts weight mode (0=icr, 1=raw weights).
- e. Falconx_listmode.def:
 - i. Added IDL codes for ICR and Raw pixel count.
- f. Fx_browse:
 - i. Add ICR, Raw and dead-time calculations.
- 9. Maia device:
 - a. Had an extra norm of 'dta' by 'maia_fixed_dwell' in 'read_buffer'.
 - b. Remove this, as it duplicates what is already done in 'da_evt' with dwell map.
 - c. Always look for segment #0 for 'get_header_info'. Use it if found.
- 10. Device documentation:
 - a. Improved notes in sources on 'dead_fraction' and its normalization to:
 - i. Dwell time (map or total) in 'da_evt', 'spec_evt'.
 - ii. Number of selected detectors in an array 'nqc'.
 - iii. In Maia, MDAQ2, DAQ36, Falconx devices.
 - iv. Fortran (event, dtfx).
- 11. Geopixe_update:
 - a. The remote CD to "Help\" fails, which uses the path separator convention of the local host. The remote (Linux) host does not like this (anymore?).
 - b. Added a 'force="/" keyword to 'fix_path()' in 'ftp_dir_copy' to force the remote site CD to be Linux/Unix based.
 - c. Version 3.6.
- 12. Version: 7.3z

4 Nov, 2016

- 1. Falconx device:
 - a. Get_header_info:
 - i. Regardless of file name passed, check for a file "*" _0.silist" first to look for JSON header.
 - ii. Try to form the _0 file as output guide for YLUT too.
 - b. Read_setup:
 - i. Needs to read a header even if 'first' is not set, in case JSON is present.
- 2. Spectrum_load_do:
 - a. If "F" is multi-file (drag select), then just use these files, and do not call 'select_evt_files'.
- 3. Spectrum_load_prep:
 - a. Set "output" path first, and pass this to flux_select' to set YLUT path correctly.
- 4. Corr: make_corr_tvb:
 - a. Use 'xanes_stack_test' and 'image_flux_charge' as in 'image_routines': 'analyze_image'.
- 5. Spec_evt, spec-xstep_evt:
 - a. Add "detector_select" argument to select the one region mask to use for /by_detector sorts.
- 6. Image_table_evt:
 - a. Use "detector_select=(*pstate).sel.top" to set selected region for individual spectra extract.
- 7. Fx_browse:
 - a. Fix incorrect sign extend on timestamps.
- 8. MDAQ2 browse:
 - a. Fixed "run" text in widget. Still shows too many files ...
- 9. FalconX device:
 - a. Use all generic counters for flux (FC0 – FC3), and selected one in flux[*,* ,0] array, but only gather FC0, FC1 as extra image planes.
 - b. FalconX_events3:
 - i. Catch any of FC0-FC3 by flux_mode (1-4) into flux[*,* ,0].
 - c. Read_falconx_segments:
 - i. Did not copy back 'versioni' with JSON version. Fixed.
 - d. Read/ write of options:

- i. Confusion around 'version' between SU listmode version and the output file version.
- ii. Added code to work-around this if the 'version' read is 803 instead of the small negative number expected (-1 up until now).
- iii. Now move to a new version of -2, which has both versions in the file.

26 Oct, 2016

1. MDAQ2 device:
 - a. Added Busy MSB record (type=3) to 'mdaq2_listmode.def' and to the Fortran device.
 - b. Adds an extra Fx entry for busy from MSB, so make sure process all as ADD. Yes.
 - c. A flaw in the MDAQ2 processing is accumulating DT per pass (if the scan passes each pixel more than once) not per event, but just per pixel. All selected detectors are added together, and so a scaling of 1/nqc is applied. The problem is if the second pass through a pixel has a very different count rate, then we should be using some count-rate weighted DT sum, not the simple sum.
 - d.
2. Fit_results:
 - a. Do not start with "veto light shallow elements" enabled by default.
3. Pixe_initial:
 - a. Found that a combination of factors in 'pixe_initial' conspire to remove Mo K lines. Firstly, the area estimate uses the minimum of the area above background divided by intensity as an estimate of area. In this case, it used the weak Ge escape peaks, which were ~below background and forced a minimum (3) area estimate. Then the 'weed out weak lines' section threw all but the first couple of lines out (all minor beta and all escape lines).
 - b. Fixed this by adjusting some parameters:
 - i. The rel used for main line tests for area estimate changed from rel = 0.003 to 0.01, and
 - ii. The minimum area of a line changed from 0.3 to 0.1 counts.
 - c. Cannot use "mean()" of area estimates, instead of minimum as lots of minor lines overlapping with major elements tend to set high initial areas.
4. EVT:
 - a. Do not set "detector array" now if n_det > 1 on Template.
 - b. This will need a device default setting/attribute later.
 - c. Set the "detector array" if the device droplist changes to such a device.
 - d. Use new base device method Obj->array_default() to test whether a device supports detector arrays by default. Use this in 'evt_set_device' to set array mode.
5. Devices that support detector arrays by default:
 - a. Add the init "array_default=1" to each and re-compile.
 - b. APS, Elettra HDF, GSE-CARS MCA, HI-PIXE iXCCMap, HS-PIXE HDF, Maia, NSLS HDF, NSLS MCA, NSLS NetCDF, PNC-CAT HDF, Primecore U48, Sandia BD12.
6. MM Camera View:
 - a. Added tempo code to read Gareth's finder mosaic image.
 - b. Uses temp file for PNG binary, and the 'image_read()'.
 - c. Still needs to conform to KVS definition.
7. Falconx device:
 - a. Fx_browse:
 - i. Fix file names and runs.
 - b. falconx_listmode.def
 - i. Added a list of 'normal_accept' to restrict tags to usual ones.
 - c. Falconx_version
 - i. Make it immune to an INT passed as 'json'
 - d. Read_falconx_json
 - i. Make it able to cope with no header and a null header and no JSON.
 - e. Read_falconx:
 - i. Pass 'accept' to filter tags.

- ii. Take care not to include words not in the current tag (if 'skip' is used to jump to some point_lun).
 - iii. Default to just 50 kb
 - iv. Allow no records returned from 'read_falconx' to be legal, as we are using a cut-down 'accept' list, and so sometimes only get 'sync' events in first 50,000 bytes.
- f. Read_falconx_segments:
 - i. Default to just 50 kb.
 - ii. Read 'json' and get 'version' and 'njson' and pass back.
- g. Read_falconx_header:
 - i. Get 'json', 'njson', 'version' from 'read_falconx_segments' and pass to 'get_falconx_details' to output in head 'd' struct.
- h. Get_falconx_details:
 - i. Pass on input 'njson', 'version' to output in head 'd' struct.
- i. Get_header_info:
 - i. Moved the 'version' and 'njson' in self.sort_options to 'update_header_info'. This way it gets executed in 'read_setup'.
- j. Read_setup:
 - i. Enable the header read code, including 'update_header_info', and do a "point_lun, njson" to skip past the JSON header if First.
- 8. Geopixe_update, maia_update:
 - a. ftp_connect:
 - i. ftp_get_file:
 - 1. Skip out if file length is zero.
- 9. Version 7.3y

21 Oct, 2016

- 1. Moved GeoPIXE-source2 under SVN control.
 - a. Updated latest archive on Orange disk (GeoPIXE-archive/8.5/Oct-2016) to this version BEFORE doing the re-organization for SVN.
 - b. Changed organization to move out certain things (to "GeoPIXE-common"):
 - i. Workshop notes and Powerpoint files.
 - ii. Info
 - 1. Now hold all discussion/ info docs (that were in "docs") not specific to a version of GeoPIXE.
 - iii. External dir tree, including Fortran files.
 - iv. Figures and PPT for manual.
 - v. IDL issues.
 - vi. Storage of detectors, filters, pcm yields, etc.
 - c. Notes:
 - i. Still have SAV files in GeoPIXE dir and SVN.
 - ii. There are still a few other binary files:
 - 1. GeoPIXE Image, bayes_back, dforrt libraries
 - iii. Exclusions list:
 - 1. *.pref, *.bat – do save templates for these.
 - 2. .* - to exclude all local Eclipse files.
 - 3. Thumbs.db.
- 2. Pointer_display:
 - a. Work around of bug in IDL that crashes when referencing p[n] of an "apparent" array of Python objects. Caused by n_elements(p) return the number of keys or attributes, rather than the number of objects.
 - b. Also seems to crash for help 'output' keyword: "Help, p, output=s".
 - c. Logged a ticket (GALE6861A6) with ESRI.
- 3. Image_table_eventcb: OnButton_Image_Table_Export:
 - a. Fixed columns out of order.
 - b. Added "> 0.0" to suppress negatives.

12 Oct, 2016

1. Adding 'facility' and 'endstation' to allow GeoPIXE to write 'newscan' and 'position' entries to the KVS to use with Maia Mapper's Scan_List.
 - a. Maia device:
 - i. Read metadata for 'prefix' and 'endstation' from metadata.datum[] entries under tag=55.
 - ii. Add these to base device header.metadata 'facility' and 'endstation' fields.
 - b. Base device:
 - i. Add header.metadata 'facility' and 'endstation' fields.
 - c. Define: 'image' and 'stack' structs:
 - i. Add 'facility' and 'endstation' fields.
 - ii. **Not spectrum define yet.**
 - d. EVT:
 - i. Add 'facility' and 'endstation' fields to *p.
 - ii. Copy 'facility' and 'endstation' fields from mp to (*p) in 'evt_check_mp'.
 - iii. Pick up 'facility' and 'endstation' fields on image read from template.
 - iv. Pass 'facility' and 'endstation' fields to da_evt, cut_evt, da_evt_stack and da_evt_tomo in 'evt_start' (**not spectra ones yet**).
 - e. Write_geopixe_image, Read_geopixe_image:
 - i. New version -56 – write/read 'facility' and 'endstation'.
 - f. Onbutton_image:
 - i. Use 'facility' and 'endstation' to form key string for KVS write of 'GP' 'newscan' and 'position'.
 - g. Maia_defaults:
 - i. Added 'default facility' and 'default endstation' to .Maia.conf files.
 - h. Maia_launch:
 - i. Use facility and endstation in maia_defaults() to add these to the DA image struct for RT images.
 - ii. Add facility and endstation to the title bar.
2. Da_evt:
 - a. Flatten to Flux0:
 - i. Why does it not get done quite regularly?
 - ii. It is limited in cases where pixels fall below 5% of the average. A significant number of pixels with very high flux can skew this average a lot, if it makes the average more than 20x good pixel flux values. These then get ignored in the 'flatten' scaling'.
3. Da_evt_tomo, da_evt_stack:
 - a. Use 'multiplicity_scale' as in 'da_evt'.
4. set_image_minmax (image_routines);
 - a. Fixed error in 'xmax' and 'ymax'.
5. Version 7.3w
6. Evt_start_increment2:
 - a. Combining average 'yields' now uses maximum, to mostly distinguish good pixels (with events) from empty ones (yield=0). See comment “;@10-16”
 - b. This error means using versions 7.3i to 7.3w (before Oct 17), in cluster mode, would cause errors in 'yield' map and hence errors in some region spectrum overlays. Typically, this causes the reconstructed overlay to be low by about 2x.
7. Image_routines:
 - a. Analyze_image:
 - i. If 'var' is displayed ((*pstate).display_mode eq 1) then just display the average 'var' along line (no error bars, mdl).
 - ii. Can load 'var' in plugin with 'yield' and plot line profile of 'yield'.
8. Fit-setup:
 - a. Fixed non-display of adjust tweek elements and lines. Using new routine 'fit_setup_set_adjust' called from do_fit, and also when widgets are changed.
9. Version 7.3x

5 Oct, 2016

1. MM Scan_List:
 - a. Set the following metadata.datum[] entries in Maia:
 - i. 0: Key = "prefix"
 1. Value = (*pstate).kvs_prefix (e.g. "MM.Mel.")
 - ii. 1: Key = "endstation"
 1. Value = string((*pm).endstation) (e.g. "1")
 - b. Use of these in 'onbutton_image' discussed later (above).
2. PNC-CAT HDF5 device:
 - a. For processing image data, it returns buffers containing the events found in a single Y line for a detector plane, then cycles through Y and then detector channel. This is very memory efficient, but may not be fast. We'll see how it goes for real data-set sizes. May be better to do chunks of Y lines, for example. Let me know how fast a typical file scan is.
 - b. There are a few things missing from the HDF file, that will help later:
 - i. cal detector channel energy calibrations (linear $E=aX+b$)
 - ii. energy beam energy
 - iii. comment comment string for this run
 - iv. IC_name ion chamber PV name (e.g. or index into "Detectors")
 - v. IC_sensitivity ion chamber preamp sensitivity (relative to 1.0 = nA/V)
 - c. Haven't used the 'autodt', which seems to just refer to SCA maps
 - d. Assumed that the "Detector" channels for ion chambers (e.g. "Preslit") are accumulated counts (from V to F) in that pixel, and not count-rates.
 - e. Uses new 'progress_file' mode of 3 and a returned 'progress_size'= ny * ndet.
3. Sort EVT:
 - a. Seem to have fixed 'creeping growth' of Sort EVT window by commenting out the 'widget_control, hourglass=0' lines. Do not understand this ...
4. Device_specific:
 - a. New 'progress_file' mode of 3, which uses 'progress_size' returned.
 - b. New keyword: 'progress_size', to set the expected number of buffer reads for progress display.
5. Da_evt, spec_evt:
 - a. Use a new 'progress_file' = 3 and an associated 'progress_size' for the total number buffers expected (ny * ndet for PNC-CAT data).
 - b. Also updated: cut_evt, da_evt_stack, da_evt_tomo, cmit_evt, cmit_xstep_evt, cut_xstep_evt, da_trav_evt, da_xstep_evt, da_xstep_evt2, da_xstep_trav_evt, spec_xstep_evt, time_amp_evt, xanes_da_evt, xanes_single_da_evt.
6. Cut_xstep_evt:
 - a. Fixed output write to include:
 - i. img.matrix.charge=1, img.mode=1+stim_mean, to match 'cut_evt'.
 - ii. This fixes the null header image read warning.
 - b. Note: there seems to be some inconsistency in old CUTs DAI files that bomb out on read. Will need to remake these: such as demo Cooke FI cuts DAI.
7. EVT:
 - a. The 'creeping window growth' issue seems to be related to widget_control, update=1, which was done routinely in 'finish' code, even if an update=0 had not happened (e.g. with the tracking events).
 - b. Commented the update=1 in 'finish' out and it seems OK now, even though the update=0/update=1 bracketed code still exists elsewhere (e.g. device mode or flux mode routines). These do not seem to cause the creeping disease.
 - c. Now re-instated update=1 in 'finish' but with a test first for update=0.
8. Version 7.3v

27 Sep, 2016

1. IDL Compile Blues:
 - a. Since 'make_image.pro' referred to 'blog_browse', it compiled into GeoPIXE.sav.

- b. Must make sure that there are NO references in any PRO file in GeoPIXE project to external project PRO file names, which would cause them incorrectly to be compiled into GeoPIXE.sav.
 - c. Call 'blog_browser' (not 'blog_browse') in GeoPIXE to force it to load the SAV file named 'blog_browser.sav'.
- 2. Depth Wizard:
 - a. Fixed bug in calculation of step-depth yields and amended help text files and Figure 1.
- 3. Browse programs:
 - a. Tag selection: if no changes made, then do not redo list scan.
 - b. Apply this to Maia, FalconX, MDAQ2 and Midas browsers.
- 4. EVT:
 - a. Evt_device:
 - i. Set default starting PV val/unit to 1,0.
 - ii. Check PV val/unit and set droplists for these too.
- 5. FalconX device:
 - a. Make a separate NMP FalconX super-class: 'falconx_nmp_device'.
 - b. Fx_browse:
 - i. Use 'skip' to skip bytes.

16 Sep, 2016

- 1. FalconX device:
 - c. Falconx_version:
 - i. Changes to bits 24-27 codes for 'types' in all 'stats' records.
 - ii. Also changes number of words for 'stats' records.
 - iii. Added detection of a version from the JSON header ("instrument.firmwaveVersion").
 - 1. Needs passing this around via 'read_falconx_segments', 'read_falconx', 'read_falconx_header', 'get_falconx_details', etc.
 - 2. Did this by adding 'version' to object's 'sort_options' in self.
 - d. Falconx_listmode.def
 - i. Now has conditional branches on 'version' in bit mask definitions, which must be defined prior to entry (or defaults to 0).
 - ii. NOTE: need to specify any Long parameter with top bit set as 'FFFFFFFF'xl (note Long 'xl'), else it gets defined as Long64.
 - 1. Is this new to IDL 8.5! No, both 7.1.1 and 6.4 do it too!
 - 2. In Maia, tended to use 'FFFFFFFF'xul explicitly.
 - e. Fx_browse:
 - i. Add version and json string to pstate.
 - ii. Recover negative pixel addresses.
 - f. Falconx_events3:
 - i. New version of the Fortran that uses the 'version' and variant formats.
 - ii. Detects negative X,Y,Z and sign extends them.
 - iii. Uses top 13 bits of E data.
- 2. Blog Browse:
 - g. Move all bit pattern masks for Blog Browse into 'maia_listmode.def'.
- 2. Widget_list index error:
 - a. Scrolling a large list (more than 64K items) returns a 'event.index' that wraps at 16 bits (unsigned).
 - b. But using 'widget_control, list, set_list_select=n' works for large 'n'.
 - c. Logged call to ESRI (G9GE35579A), Harris (IDL-69629).
- 3. **NOTE: Warning for any Device driver:**
 - a. Added 'version' to self. But 'da_evt' and 'spec_evt' do not use the original object, and pass 'devpars' to restore object state.
 - b. Hence, any new stuff, like 'version', needs to be in 'self.sort_options' and be set by 'set_options' method, so it is seen in the obj for 'da_evt', 'spec_evt' via 'devpars'.

7-15 Sep, 2016

1. MDAQ2 device:
 - a. Read_setup:
 - i. If PV is blank, define: mdaq2_IC_name='mdaq2:scaler.FC0'
 - b. Read_buffer:
 - i. Move dt line outside 'mdaq2_hw_scaler' test (for spectra).
2. Maia device:
 - a. For Maia Mapper
 - i. No flux → use Dwell time normalization
 1. New Maia PV "Maia:dwell.time"
 - a. maia_flux_mode = 3
 - b. maia_hw_scaler = 1
 2. 'maia_384_events6' needs to return BT → maia_fx[0,*] for this mode
 3. Set sensitive=0 the Preamplifier settings.
 - b. read_setup:
 - i. Make sure nsfs_flux_scale=1 if no PV is defined.
3. Maia Control:
 - a. Count-rate maximum in Rate window
 - i. Red histogram in Red behind the Green one.
 - b. Version 7.3u
4. Scan_list:
 - a. Top-Right/Bot-Left and Top/Bot sequences now work correctly.
 - b. If Presets not found, then REQ again on Home timer later.
 - c. Manual move: Check for numeric values, flag error if bad. If blank, then default to current position.
 - d. Added basic shutter controls for connected end-station.
 - e. Version: 1.4
 - f. Refined use of separate "Source" and "Target" coordinates controls.
 - g. Enabled translation from source "mapper1" to target "mapper2", for example.
 - h. Version: 1.5
5. GeoPIXE:
 - a. Fit_setup:
 - i. "Free Gain" now initializes correctly.
 - b. Multi-Image:
 - i. Fixed resizing for non-square aspect ratio images.
 - ii. Fixed notification of new images loaded.
 - c. Image 'Analyze' menu:
 - i. Added 'Types' "Spline 100" and "Single Pixel"
 - d. Python routines:
 - i. Refines kvs and vsub routines to handle kvs offline better.
 1. Use 'check_kvs' in initial 'open_kvs' call in IDL main routine, and if it fails, 'close_kvs' (which now sets return 'kvs' to 0 – not an object).
 2. Further kvs routine calls will quietly exit if the 'kvs' variable is not a Python object.
 - ii. Add /huge option to 'get_kvs' and 'set_kvs' to extend timeout (20x the default, which is 5s) for large image reads.
 - iii. Add 'query_kvs' function to use the KVS search method.
 - iv. 'close_kvs' sets kvs to 0.
 - e. Version 7.3u

19 Aug, 2016

1. Fit_setup:
 - a. If /test, then rename "Refit" button to "PyFit", which will call 'pixe_fit' with python=1 to just assemble all lines parameters, etc and export all lines data, spectrum, background, pileup, etc. and fit results to files for a Python program.

- b. Launch this by making a copy of “GeoPIXE.sav” called “geopixe_test.sav”
2. Pixe_fit:
 - a. If /python, then output files for all lines parameters, etc. and then after the fit, also output the spectrum, background and fit.
3. Maia_Launch:
 - a. Edit both project and group blog storage options.
 - b. Use new ‘blog.’ Variables.
4. Image_routines: set_image_minmax:
 - a. Went back to simpler border ignore area. The complex one involving observed ‘bounds’ ignores all good pixels when a small test subset of blog files is used.
5. Strip_clip:
 - a. Use /silent to suppress error message for zero spectrum (in Fit All loop).
6. Scan_List:
 - a. “Start” now forces a “Commit” of whole Table to the KVS.
 - b. So, that Stage daemon can read first scan details back correctly.
 - c. Do “scan_list_maia_stop” to Endrun after a scan completes.
7. Options_popup:
 - a. Add “multiple” file select vector, one for each file.
 - b. Multiple files are returned in a single string, with the full path in the first, and added files appended following "+".
 - c. To separate these into a file vector, use function: “options_multiple_files()”.
8. Peak Fit Stats Spectrum Plugin:
 - a. Use retain=2 for Linux, to keep window contents.

7 Aug, 2016

1. Continuum source:
 - a. Excise all continuum source files, which were developed jointly between MM and LAR projects, to a separate new project: “Source Continuum”.
 - b. Compile all these files into a new “continuum_source_plugin.sav”.
2. Image_table: OnButton_Image_Table_Export:
 - a. XANES mode detected looking also for all numeric column headings.
3. Scan_List:
 - a. Fixed using origin X,Y to set the pixel position in Maia scan set-up.
4. Scan_Edit:
 - a. Left/right, Top/bottom now use ‘min’ (for left, bottom) and ‘abs’ for size.
5. Maia_Rates:
 - a. Now plots max count-rate in Red (behind Green histograms).
 - b. Clear these with ”Clear” button.
6. Maia_setup:
 - a. ‘Load parameters’:
 - i. Now uses the ‘file_requester’ /skip_if_exists mechanism to provide a fall-back pop-up if files are not found, prompting with file-name.
 - ii. Uses default path same as parameters file. With /skip_if_exists above, this means it finds all files in the same dir.
7. Add “maia:dwel.time” to allowed PVs for Maia device (as mode 3):
 - a. check_plist_maia:
 - i. Add “Maia:dwel.time” to PVs for Maia.
 - ii. Treat it as another H/W scaler.
 - b. Flux_select, EVT:
 - i. Any occurrence of “.time” in (*p).preamp.pv will sensitive=0 the preamp sensitivity and units droplists.
 - c. Maia device:
 - i. Setup:
 1. Detect ‘Maia:dwel.time’ as H/W
 2. Set ‘maia_flux_mode’ = 3

3. Set 'nsls_flux_scale' =1.0
- d. Maia_384_events6:
 - i. For 'flux_mode'=3, set fx(0,n)=fx(3,n)
- e. Maia_accumulate_dtfx (and _3D):
 - i. Only tests for 'flux_mode' not 0, so no change needed.
8. Version 7.3t

22 July, 2016

1. Onbutton_image:
 - a. Use returned 'sunits' for single pixel distance units on 'UP'.
2. ScanList:
 - a. Chasing a lock-up in MM Python libs and ZMQ, now use separate 'comm' objects for KVS, and Stage and Security VSUBs.

30 June, 2016

1. Scan List:
 - a. Scan Sequence parameters (in *pstate):

i. scan_sequence.active	flags scan sequence on
ii. scan_sequence.scan.index	index in *pindex list
iii. scan_sequence.raster.on	flags raster on
iv. scan_sequence.raster.pause	flags raster paused
v. scan_sequence.raster.error	flags error returned from stage
vi. scan_sequence.time_start	time since start of scan
 - b. Scan Sequence logic:
 - i. After a scan has been started, a 3s delay is used before acting on any "raster" status returns from the Stage. This ensures that the "raster" state is up to date and not stale.
 - ii. Then "raster".status = "stopped" is looked for to indicate that the scan has ended.
 - iii. Go to next scan if active=1 and raster.on=0 and raster.error=0
 - c. Next scan:
 - i. If previous scan was active = STOP, or no more ON, then completed.
 - ii. Index down list using *pindex
 - iii. Look for next scan with active ON, START or STOP.
 - iv. Save index as 'scan_sequence.scan.index'
 - v. Send VSUB the "Execute-Scan" REQ.
 - d. Stage daemon needs:
 - i. Must update "raster" status a few times per 3 seconds.
2. Python:
 - a. The following are needed in GeoPIXE installation and system paths:
 - i. Windows:
 1. Patches to idlrt.exe, idlde.exe, idl_...
 2. See "IDL Python Error R6034.docx" in "doc\IDL Issues".
 - ii. PATH as needed to run Python (Anaconda2 4.0.0 → Python 2.7.11)
 - iii. Python modules:
 1. MM libraries
 2. My Python Scripts
 - iv. PYTHONPATH to include MM libs and My python scripts.
3. DAQ:
 - a. Build_raster_list4:
 - i. Clear pixel.dim[].enable for axis 4.
 - ii. Add beam.photon.enable, beam.enable
 - iii. Make sure the last 'stop' step has move.beam.enable, dwell.beam.enable both set.
 - b. DAQ_launch_init:
 - i. Add beam.photon.enable, beam.enable
 - c. Daq_launch on stop:
 - i. Make sure the last 'stop' step has move.beam.enable, dwell.beam.enable both set.

13 May, 2016

1. Bake-out definitions changed:
 - a. Bake-out plug inserted enables either Cooling or Bake-out.
 - i. Bake-out plug inserted is given by `(*pm).controls.status.bake`
 - ii. Bake-out in action now measured by `((*pm).controls.peltier lt -0.02)`
 - b. Add a new droplist to Maia-setup to select Cool or Bake
 - i. The latter is only available with plug inserted.
 - ii. New variable `(*pstate).peltier_mode`, which reflects droplist.
 - c. Maia_setup:
 - i. Maia_setup_check_warning:
 1. `bake_on` uses `((*pm).control.peltier lt -0.02)`
 - ii. Maia_setup_update_dynamic:
 1. Set `(*pstate).peltier_mode` and Cool/Heat droplist
 2. Based on `((*pm).control.peltier lt -0.02)`
 - iii. OnRealize_maia_setup_peltier_mode:
 1. Set `(*pstate).peltier_mode` and Cool/Heat droplist
 2. Based on `((*pm).control.peltier lt -0.02)`
 - iv. Widget definition:
 1. Add a new 'controls_peltier_mode' droplist, pstate entry.
 2. And a mode variable in pstate: `(*pstate).peltier_mode`.
 3. Make Bake warning smaller.
 - v. Event routine:
 1. Peltier-mode:
 - a. Only allow Cool, unless Bake-oput plug installed.
 - b. If reverse to peltier current, zero current.
 2. Peltier-slider:
 - a. Cool/Heat state now given by `(*pstate).peltier_mode`
 - d. Maia_update_parameters:
 - i. `Led[5]` set to `((*pm).control.peltier lt -0.02)`
2. Interlock alarm:
 - a. Maia_launch:
 - i. If interlock fails, pop-up warning: 'alarm_popup', only if popup is not already up and flashing.
 - ii. Use new detector bias min/max variables (names changed).
 - b. Maia_update_parameters2:
 - i. `Led[4]` set to `(*pm).control.interlock`
 - c. State_button:
 - i. Added 'charthick' keyword.
 - d. Alarm_popup:
 - i. Red/yellow flashing warning in centre of screen.
 - ii. Uses 'state_button' with large charsize and charthick.
3. Version 7.3q

10 May, 2016

1. Maia_launch:
 - a. Maia_launch_version2:
 - i. Use new detector bias min/max variables (names changed).
2. Maia_setup:
 - a. Use new bias min/max for slider range.
3. Define:
 - a. Add 'bias_min' and 'bias_rate' to 'control' in maia_struct.
4. Image_history:
 - a. Double click on pixel statistics lines to get a frequency histogram of dwell, count-rates, pile-up and flux.
5. Gas absorbers/ filters:

- i. Goal to always store gas absorber layers at mg/cm^2 assuming NPT pressure (1 atm; 1013.25 mbar) and temperature (20 °C).
 - b. Filter_setup, source_setup, detector_setup:
 - i. Add a new thickness mode “Gas (mm NPT)”.
 1. Detect density less than 0.01 as indicating a Gas layer, set mode accordingly, and convert microns to mm thickness.
 2. Reverse this on save.
 3. If thick or formula changed, calculate density.
 - ii. ‘xxx_update_density’ routines:
 1. Reinterpret ‘formula’ to get weight of gas molecules
 2. Calculate density for ideal gas.
 - iii. On load (for a gas):
 1. If density < 0.01 then set thickness mode=2.
 2. If thickness mode =2 scale thick / 1000 to mm.
 - iv. On save (for a gas):
 1. Clip micron thickness mode to 1
 2. If thickness mode =2 scale thick * 1000 to microns.
 - v. Changes to low-level ‘filter’ definitions and on disk formats:
 1. None.
 - vi. For gas, only allow a “plain” filter special/pinhole mode.
6. Pressure and Temperature in absorbers:

(see discussion in “Transmission and PT variation.docx”)

 - a. Transmit:
 - i. ‘pressure’ (mbar) and ‘temp’ (C) key-words.
 - ii. Ignored if P=0. T=0 is accepted.
 - iii. Scales ‘thick’ assumed to be NPT conditions.
 - iv. Normal filters only, not pinholes, Bragg, etc.
 - b. Source_tube_spectrum:
 - i. ‘pressure’ (mbar) and ‘temp’ (C) key-words.
 - ii. Pass these to ‘transmit’ for filters.
 - c. Source_calculate:
 - i. ‘pressure’ (mbar) and ‘temp’ (C) key-words.
 - ii. Default to NPT (1013.25 mbar, 20 C).
 - iii. Pass these to ‘Source_tube_spectrum’ for filters.
 - iv. /convert to convert local (source_setup) to standard mg/cm^2 thick.
 - d. Det_eff:
 - i. ‘pressure’ (mbar) and ‘temp’ (C) key-words.
 - ii. Pass these to ‘transmit’ for internal/external filters.
 - e. Array_yield:
 - i. ‘pressure’ (mbar) and ‘temp’ (C) key-words.
 - ii. Pass these to ‘det_eff’ and ‘transmit’ for array intensity calc.
 - f. Pixe_fit:
 - i. ‘pressure’ (mbar) and ‘temp’ (C) key-words.
 - ii. Pass these to ‘array_yield’ for array intensity calc.
 - g. Fit_setup:
 - i. If ambient conditions change (ambient.on not zero, changed from previous calculation), then call ‘fit_recalculate_yields’, which is based on yield calculation in ‘yield_setup’.
 - ii. Save new yields struct to use in fit, and save new ambient struct.
 - iii. Clear (*p).ambient.on if load new PCM or after yields calculation Notify.
 - h. Spectrum_History:
 - i. Add ‘ambient’ and ‘tube’ parameters to display list.
 - ii. Respond after Notify to work out ‘current_history_display’.

5 May, 2016

1. Set_image_view:
 - a. Cap zoom clicks to 4, or 2 above 1M pixel, or 0 above 25M pixel.
2. Flux_flatten:
 - a. Change 'base' threshold to 5% instead of 10%.
3. Analyze_image:
 - a. Guard against an MPDA image, but failing to find the matrix.
 - b. Then (*p).has_yield=1, but 'matrix' is not a struct possibly.
 - c. Have 'use_yield', which equals (*p).has_yield, except when matrix not found.

20 Apr, 2016

1. DAQ_launch:
 - a. Handle all new variable names (assumes svn > 7002 now).
 - i. Get_daq_details:
 1. From var_val:
 - a. Metadata: sample name, type; charge coeff, unit
 - b. position.dim[].name
 - c. scratch.datum[]: 19: deadline
 2. From metadata:
 - a. Beam energy; sample info (comment) and scan region (grain) and order
 - b. scan axis details: name, origin, extent, pitch, enable
 - c. charge: coeff, unit
 - ii. daq_device:: get_header_info:
 1. Checks scan.order and sets device axis.x,y
 - iii. daq_device:: update_header_info:
 1. Takes details and fills in 'header' details for:
 - a. Scan parameters, sample details, metadata, IC, energy, deadline.
 - b. Build_raster_list4:
 - i. Do bounds checking on stage and DAC axes before building raster.
 - ii. This needs to account for deflect 'scale', which affects DAC range.
 - iii. Write 'metadata.scan.crossref'.
 - c. Stage_list_check_bounds:
 - i. Move all bounds checking into this routine.
 - ii. Use /silent to suppress warnings (use them only w/ "Check" button).
 - d. daq_launch_scan_request:
 - i. Readback current blog number and add to previous scan.
 - ii. Process "REF:n" in "scan-new" scratch command.
 - iii. Trigger bounds checking and "yellow" in List if bad.
 - e. Stage_List:
 - i. Add 'blog' and 'crossref' columns, linked to (*pm).run.number and "REF:n" coming in via "scan-new" command to scratch.
 - ii. Rows with bad bounds results show as yellow.
 - f. Stage_Edit:
 - i. Add deflect DAC scale variables, sensitive=0 when XY not DAC.
 - ii. Add "Check" button ("yellow" when bounds bad) to probe bounds.
 1. Will pop-up error warnings for specific bounds failures.
 - iii. Moved "Beam" to a pointer 'pbeam' shared between DAQ_Launch, Stage_List and Stage_Edit.
2. DAQ_setup:
 - a. Deadline parameters and "auto" mode:
 - i. daq_setup:
 1. Added "auto" checkbox to Device tab, and an "Apply" button.
 - ii. daq_setup_apply_device:

1. Get `deadtime_cal` from device, package in string and send to klee to store in `'scratch.datum[19].value'`.
2. New struct string: `"{auto:1,cal:{a:10.,b:0.}}"`
3. Copy `(*pl).deadtime.cal` to local `*pm`.
- iii. `daq_client_parameters`:
 1. Reads Klee `scratch.datum[19]` and copies string into byte array `(*pm).info.deadtime` in shared memory (note this is shared version of `*pm`, which has a different form and no strings).
- iv. `daq_launch_read_info`:
 1. Called from `'daq_launch_read_enable'`, which is called from `'daq_update_parameters'`.
 2. Reads byte array in shared memory and writes into `*pm`.
- v. `daq_launch_update_deadtime`:
 1. Checks "auto" mode and sets DT cal A and writes back to Klee if need be.
3. DAQ 36 device:
 - a. Fortran: `daq_36_events`
 - i. 'skip' was still INT, which caused negative skips when skipping the new long `Var_val` records; changed to LONG. This has been fixed in Maia device, but not DAQ 36.
 - ii. Fixed `daq_36_events.pro` as well to check for LONG skip now.
 - iii. Added test for negative skip, and new idebug return.
 - b. `Get_daq_details`:
 - i. Handles all new metadata and `var_val` variables.
 - c. `Update_header_info`:
 - i. Assigns scan axis details re-directed via `'axis.x'` and `'axis.y'`
 - ii. Uses new metadata.
 - d. `Get_header_info`:
 - i. Decodes the "scan.order" metadata to set `axis.x`, `axis.y`.
 - ii. Done here, not in `'update_header_info'`, to allow changing axes.
 - e. New method `'update_device_from_header'`:
 - i. Update any device `sort_option` parameters found in recently read header.
 - f. `daq_accumulate_dtfx`:
 - i. A clone of `'maia_accumulate_dtfx'`. However, it uses a new parameter `'dte'` as DT cal for the NIM channels.
 - g. Read-buffer:
 - i. Ignore `HW=0` (`time_last`, `dt`, ...), as this only happens via Time-Amplitude sort.
 - ii. Use INT positions 0,1,2,3,4,5, but then convert to UINT for `'daq_accumulate_dtfx'` and return to GeoPIXE as usual, following already done in MDAQ2 device.
 - iii. Comment out scaling NIM T to use same cal as Scepter channels.
 - h. Deadtime Cal parameters from header:
 - i. Need to update Deadtime Cal in device `'sort_options'` as read in `'header'`.
 - ii. This is now done in a new method `'update_device_from_header'` (in both DAQ and Maia devices, with default no-op in `base_device`).
 - iii. Called in `'spectrum_load_prep'` for *Spectrum Import* and in `'evt_check_mp'` in EVT for blog files in *Sort EVT* window.
 - i. Deadtime Cal and effect on T and dead-time estimation:
 - i. Since `DTcalA` differs from `DTcalNIM`, we scale T for NIM channels so they can use the same `DTcalA` as the SCEPTER channels.
 - ii. So: $T * \text{deadtime_cal_nim} == \text{newT} * \text{deadtime_cal_a}$, which means:

$$\text{newT} = T * (\text{deadtime_cal_nim} / \text{deadtime_cal_a})$$
 - iii. This is done in `'read_buffer'` after `'daq_accumulate_dtfx'`, which already knows about the difference.
 - iv. This process does produce some aliasing in the T spectrum for NIM channels, which should not be an issue.
 - j. YLUT and the 'slow axis':

- i. No slow axis controls yet, and YLUT is NOT enabled in device 'init'.
 - ii. May not be an issue, because we probably won't need cluster mode for DAQ device.
- 4. Time-Amplitude window:
 - a. Added menu for DAQ 36 device, and now pass device name to extract routine.
 - b. New quirk of IDL 8.5.1:
 - i. It calls the resize event routine immediately after a plot draw for some reason (even before the draw routine finishes).
 - ii. Needed to add a new variable 'skip_resize' to pstate to stop the plot forcing an immediate resize event (before draw routine actually finishes). Resize will ignore these events until draw has finished and flag cleared.
- 5. Maia example as basis for DAQ launch:
 - a. How are DT cal parameters set "auto" in Maia?
 - i. Where to keep DT parameters in Klee?
 - 1. Add DTcalA, DTcalB to auto in info struct in scratch.
 - 2. Add to 'get_daq_details' to decode this from var_val.
 - ii. Maia_launch_read_info
 - 1. Reads deadtime.auto from shared memory (if avail.) or from 'scratch.datum[].value into (*pm).deadtime.auto
 - 2. Called from 'maia_launch_read_enable'
 - iii. maia_launch_read_da:
 - 1. Reads deadtime cal and imaging parameters from shared memory and writes *pm
 - 2. Invoked:
 - a. Rates timer: if auto-save DA image, or new run.
 - b. maia_launch_initial
 - c. maia_setup_apply_imaging
 - d. maia_update_da2
 - iv. maia_launch_update_deadtime:
 - 1. Checks 'auto' and if set updates DTcalA, DTcalB.
 - v. Maia_update_parameters:
 - 1. Calls 'maia_launch_update_deadtime' to handle auto
 - vi. Maia_setup:
 - 1. Needs a "auto" checkbox on HYMOD tab, with 'hymod-deadtime-auto' event to set (*pl).deadtime.auto
 - 2. Initial set-up of 'hymod_deadtime_slope' widget mentions auto.
 - 3. New 'hymod_deadtime_auto' check-box.
 - 4. Add 'hymod_deadtime_auto' widget to state
 - vii. Maia_setup_apply_hymod:
 - 1. Sets the scratch.datum for deadtime info including 'auto'.
 - viii. maia_setup_update_dynamic:
 - 1. Sets checkbox based on (*pm).deadtime.auto
 - ix. Maia_client_parameters:
 - 1. Reads deadtime info from scratch into (*pm).info.deadtime shared memory.
 - x. Maia_client_parameters:
 - 1. Reads deadtime cal parameters
 - a. These do not exist in DAQ, so will need to use scratch for them.
- 6. Maia device:
 - a. Added the new method 'update_device_from_header' to update any device sort_option parameters found in recently read header.
- 7. Base device:
 - a. Added a 'no op' for the new method 'update_device_from_header'.
- 8. File Requester:
 - a. Fixed a bug that caused some /skip_if_exists situations to return incorrectly.
- 9. Image_table_event: EVT
 - a. Fixed a bug that caused some blank final blog file names to cancel sort.

15 Apr, 2016

1. Shared memory for parallel processing:
 - a. Using one-off shared memory with unique names.
 - b. Names use 'user_#_day_hour_min_sec_' as a prefix (# = process index)
 - c. Need to shorten these names for memory segments.
2. Parallel_client:
 - a. Assumes all shared memory values set-up afresh and shared memory requested with /init, and passes the memory segment 'prefix' name to the background process.
 - b. Limitation: It uses 'common', which means it cannot be used again for the same process (e.g. no "EVT" as the same time as a running "Image").
 - c. Advantage: As many instances of GeoPIXE can be run with cluster processing in parallel.
3. Geopixe_parallel:
 - a. Retrieves the memory segment name prefix from argv[2].
 - b. Now uses gprint' for debug lines.
4. Shared_memory_buffers:
 - a. Now uses gprint' for debug lines.

11 Apr, 2016

1. Calc_slices2 bug:
 - a. Does not calculate slices in 3-layer (fluid inclusion) mode correctly, so yields can be wrong. Fixed 2 "break" → 'continue' statements.
2. Maia Control:
 - a. Socket_command_set:
 - i. Workaround a Kandinsky issue (does not respect embedded "" string).
 - ii. Now replaces embedded double " with single ' and encloses whole string in double "", which is then respected by Kandinsky.

8 Apr, 2016

1. Shared memory (for parallel processing in GeoPIXE):
 - a. Windows:
 - i. Shared memory remains once it is mapped until all processes die.
 - ii. 'shmmap' simply connects to it if it already exists, no errors.
 - iii. Need to test shared memory (e.g. ppar) contents to see if it is new or pre-existing.
 - iv. Test a fixed parameter (e.g. buffer size) or a magic number.
 - b. Linux:
 - i. Shared memory remains once it is mapped even after all processes die.
 - ii. 'shmmap' throws an error if it already exists. However, if you continue with 'shmvar' it maps fine.
 - iii. Have used 'shmmap' error to indicate shared memory already exists.
 - iv. Better to test shared memory (e.g. ppar) contents to see if it is set-up correctly, i.e. pre-existing.
 - c. Issues addressed:
 - i. Need to make sure that if multiple foreground sessions try to use the same shared memory that some checks prevent chaos. This includes the one foreground GeoPIXE session trying EVT Spectra while Images are being sorted, for example.
 - ii. This does tie up the shared memory for a long time. A better approach may be needed (see "notes for later" below).
 - d. Cluster_client:
 - i. Launch:
 1. Parallel_config:
 - a. If OK, then 'parallel_init'.
 2. Parallel_start:
 - a. If error then 'parallel_release', 'parallel_cancel'.

- b. Note: **This ‘extra’ parallel_release may ‘break’ CWS usage?**
 - ii. Event loop:
 - 1. Parallel_progress:
 - a. Uses average of all (*pf)[0] to determine fraction done.
 - 2. Parallel_monitor:
 - a. Returns clstat (>0 success, <0 error)
 - b. Parallel_results: If success.
 - 3. Parallel_release
 - 4. Parallel_cancel
- e. Parallel_client:
 - i. Config:
 - 1. Tests pointers to shared memory in common. If OK, just use them.
 - 2. Tests (*ppar)[1] as a shared memory “busy” indicator. If busy popup a warning and exit.
 - 3. Loop through all nodes to make sure all are setup before testing (*ppar)[1].
 - ii. Init:
 - 1. Spawn background processes, link to logical units to enable closure.
 - 2. Use /noshell for Linux, in which case command and args must be a string vector.
 - iii. Start:
 - 1. **NOTE: No modification to any shared memory before here!**
 - 2. If we get here, “busy” must not be set, so initialize memory.
 - 3. Sets (*ppar)[1]=1 to “busy”.
 - 4. Clear (*ppar)[2] to indicate wait for command string.
 - 5. Upload command data.
 - 6. Sets (*ppar)[2]=1 to indicate command data is ready.
 - 7. Return good status, even if command string is null.
 - iv. Monitor:
 - 1. Checks for (*ppar)[4]=1 (all processes) to indicate all done.
 - v. Results:
 - 1. Reads back all result data.
 - 2. Clears (*ppar)[1] to indicate shared memory not “busy”.
 - vi. Release:
 - 1. Sets (*ppar)[3]=1 kill flags to stop backgrpund process.
 - 2. Clears (*ppar)[1] to indicate shared memory not “busy”.
 - vii. Cancel:
 - 1. Closes logical units connected to backgrpund processes to kill them.
- f. Geopixe_parallel:
 - i. Sets (*ppar)[5]=1 to indicate up and running.
 - ii. Waits for (*ppar)[2] to be set to indicate command data loaded.
 - iii. Clears (*ppar)[2] once data has been read.
 - iv. Sets (*ppar)[4]=1 to indicate results have been uploaded.
 - v. Clears (*pb1)[0] on exit, if null command string.
 - vi. Clears (*ppar)[5] on exit.
- g. Shared_memory_buffers:
 - i. **NOTE: Must not set pars if shared memory already setup; just connect.**
 - ii. Use (*ppar)[0] (n_buffers) and (*ppar)[13] (buffer_size[0]) as indicators of setup. If they’re not OK, then set all *ppar.
- h. Shared_memory_struct:
 - i. **NOTE: Must not set pars if shared memory already setup; just connect.**
 - ii. Use (*ppar)[6] as a magic number to test. If it’s not OK, then clear all *ppar and also set the magic number.
- i. Notes for later:
 - i. If we were to separate the **client (command) → server** and **server (results) → client** data transfers into quite separate shared memory segments, then we could:

1. Only set the busy (*ppar)[1] flag during the command transfer, for the command shared memory, and only set the busy (*ppar)[1] flag during the results transfer, for the results shared memory.
2. This way they are only “busy” for the short transfer periods.
2. DAQ Control:
 - a. Sub-processes:
 - i. Launch in ‘spawn_daq_clients’ and ‘spawn_klee_clients’:
 1. Use the same mode as on Maia, with ‘spawn, /noshell’ and a vector of args.
 - ii. DAQ launch: Now also sets shutdown par to kill ‘daq_client_parameters’ properly.
 - b. Daq Launch:
 - i. Differentiate Klee versions and parameter changes:
 1. ‘position.dim[].position’ only after version 6260
 2. ‘pixel.dim[].coord’ only after version 6880
 3. ‘raster.status’ only after version 6260
 4. ‘raster.status’ → ‘raster.status.state’ after 6880.
 5. Affects: ‘daq_launch’, ‘daq_client_parameters’, ‘stage_list’.
 - c. Daq_defaults:
 - i. Add daq.device to the daq_defaults struct.
 1. Use this in Daq_launch now instead of hardwired device.
 2. Reads ‘daq device DAQ_DEVICE’ from conf files.
3. Maia Control:
 - a. Maia_defaults:
 - i. Add maia.device to the maia_defaults struct.
 1. Use this in maia_launch now instead of hardwired device.
 2. This will enable adding ‘MAIA_NMP_DEVICE’ to conf files.
 - b. Spawn_maia_clients:
 - i. Fixed incorrect form for ‘args’ under Unix. Now a string vector.
4. GeoPIXE:
 - a. Geopixe_Parallel:
 - i. Parallel processing log files now include the date-time to make them unique.

2-6 Apr, 2016

1. Composition loop in fit:
 - a. Fit_setup:
 - i. New widgets:
 1. New “multiphase loop” tab to select Mode between single or multi-loop modes.
 2. New fields for .comat file, New, Load, where “New” brings up correct_yields with mode=0.
 3. Duplicate Yields from main Setup tab to this tab as well, in parallel (need to set_widget_text to yield_file, yield_file2).
 - ii. New routines:
 1. fit_setup_elements:
 - a. build element list from *p Z, Shell (where shell=3 or 4,5 has special meaning depending on use_m).
 2. fit_setup_trim_list:
 - a. Calls ‘select_element_lines’ to trim element list and all arrays to selected elements.
 3. Fit_setup_phase_yields:
 - a. Use R matrix to transform ‘conc’ to phase fraction.
 4. Fit_setup_merge_yields:
 - a. First checks all file paths for end-members.
 - b. Using existing (*p).peaks as a template, it reads all yield files and forms weighted phase average of *p ‘yield’ and ‘intensity’.
 - c. Yields use phase weighted 1/Y.

- d. Yield per line is effectively $Y \times \text{intensity}$. $1/Y$ approach suggests each line $1/(Y \times \text{intensity})$ should be phase weighted similarly, and then 'intensity' per line = $1/(\sum \text{phase}/(Y \times \text{intensity}))$.
 - e. Renormalize lines to major line index 0 afterwards.
 - f. Any none Finite() (Nan or Inf) intensity are set to zero.
 - g. Copies final yields back into (*p).peaks
 - iii. Load/ Save PCM files:
 1. Include mpda_mode in PCM file (version = -11)
 2. If mpda_mode ge 1, then save/load correct_file string too.
 - b. Pixe_fit:
 - i. Multiphase calculations:
 1. If loop count is greater than one, it calls 'fit_setup_phase_yields' to form weighted yields/intensities based on phase fractions determined from current 'conc'.
 2. Copies these back into (*p).peaks, in 'fit_setup_merge_yields'.
 3. Loops back to start for another iteration of fit, with /use_last set to re-use 'array_yields' results for detector array variation.
 4. (*p).peaks.yields end up in results struct.
 5. Add new "multiphase" struct to fit results with mode, correct filename, minerals and phase fractions.
 - c. Correct_yields:
 - i. Now supports both Composition matrix (.comat) and Image Correction (.correct) functions. The former reads .yield files, the latter .dam* files.
 - ii. Have "Import" function to cross-load these between instances.
 - iii. Add a timer event to reset any scroll, using set_table_view=[0,0], as the scroll=0 creation option does not work (on Windows?).
 - d. Make_peaks:
 - i. Accepts 'default' as a pointer or struct to avoid new memory allocation or large argument passing.
 - e. Free_spectrum_state:
 - i. Added 'pcorrect' and recent ptrs (save_detector, playout, Compton, xanes_eneegies) to list to be freed.
2. Sort EVT:
- a. Now has "Multiphase DA (MPDA)" option in projection modes on the DA tab, which offers files with filter "*.mpdam".
 - i. Needed to update the 'file_ext' and 'file_title' arrays in 'evt_event' and in 'evt_start'.
 - ii. 'evt_set_proj_mode' needed an added 'MPDA' mode.
 - iii. Event for mode needed 'MPDA' added to options list.
 - iv. Added to 'project_modes' array in widget setup routine.
 - v. Now sets 'img.mode' to 3 in 'da_evt' for MPDA mode.
 - b. "New" button:
 - i. Pops up dialog ('options_popup') to collect phases, correct filenames and create a new (binary /XDR) .mpdam file.
 - c. Read_mpdam:
 - i. Now can read new /XDR binary .mpdam or old text file version.
3. Correcting file paths, MPDA and cluster mode:
- a. Issues:
 - i. Make sure all files referenced in *Sort EVT* window can be found before sending command to 'da_evt' or background process.
 - ii. Most files may be found somewhere on the output analysis 'path' where the input "From DAI" template may have been loaded from. But the input raw data 'dpath' might need to be selected manually.
 - iii. MPDA is special as more files are referenced in the .mpdam file (phase DAI and .correct file) and even more from the .correct file (DA matrix files). For this reason

cannot just send .mpdam file-name to background process as the files in it may not be found.

b. Solutions:

i. Evt_start:

1. Make sure all files can be accessed before sending them on.
2. If it can't find them locally (up_dir=3, /translate, alt path), it pops up file_requester().
3. Any that are found in a new path are updated in *p struct and the *Sort EVT* window.
4. For a DA file that is a .mpdam, 'evt_start' calls 'read_mpdam', which also checks file paths and finds them, else popping up file_requester(). It then uses 'stringify' to stringify the entire mpda struct returned (which contains all modified paths) to make an alternate 'mpdam_string' argument to 'da_evt'.

ii. Da_evt:

1. Has an extra keyword 'mpdam_string' for stringified MPDA struct. If the 'matrix_file' extension is .mpdam it then checks for the 'mpdam_string' keyword and passes this to 'read_da' instead of the usual 'matrix_file' filename.

iii. Read_da:

1. Checks the input 'file' for a leading "{ character indicating a stringified structure. If found, it uses 'unstringify' to form the original MPDA struct, else it calls 'read_mpdam' to read it.

4. Delete temp .spec.0, .spec.1, .spec.2 files

a. Write_spec:

- i. Search for and delete file (which includes .spec ext).* files.

5. Incorrect path root in Sort EVT formed by 'build_output_path':

- i. This caused subsequent output filenames to get crazy extra characters and longer length.

b. Evt_set_evt_file:

- i. Needed to call 'build_output_path' with /set to update root correctly.

6. Version 7.3k

29 Mar, 2016

1. Problem with fixed Width not being kept for next fit. (see “;@29-3-16”)

a. Pixe_cleanup:

- i. Always set '(*p).detector.w0' and '(*p).detector.w1' if masks are on. So that we 'remember' peak width to refit extended range with mask off.

b. Fit_setup:

- i. When spectrum changes (on notify 'spectrum-fit'), reload width w0,w1 in (*p).detector from (*p).save_detector, and set Width widget.
- ii. Also do this in “Fit All” loop mode in 'fit_setup_do_fit' for all spectra, if width is Free.

26 Mar, 2016

2. MDAQ2 device:

a. Offsets:

- i. Make start_adc=0, which means that channel number is the same in raw data as displayed (may not use #0).
 1. Start_adc=1 had the strange behaviour that it assumed channels 1,2 occurred in files as 0,1 (e.g. as old mpsys). This is not what we want in this case.

b. Get_mdaq2:

- i. Need to read calcs in to cal[] in same index as read channel #.
- ii. Also read “st0” in case Jamie wants to use it.
- iii. Will set 'detector type' (if answer “yes” to calcs).

c. Raw data and Fortran:

- i. Process ADC data, busy count, position, spatial stats.
 - ii. Fortran now has 'clock' input for FPGA clock (e.g. 80 MHz).
 - iii. Dead-fraction and dwell maps accumulated in ms time.
 - iv. In image mode, dead_fraction divided by 'nc' in array mode.
 - v. In spectrum mode, total dwell accumulated in ms too.
- 3. Maia NMP device:
 - a. Use the Maia device as a super class, and change the parameters:
 - i. Ionbeam=1, synchrotron=0, title.
 - b. Chase all occurrences of "MAIA_DEVICE"
 - i. Use DevObj where appropriate, or leave as the default device.
- 4. Maia device:
 - a. Now does NOT set ionbeam parameter.
- 5. Maia_Setup:
 - If "ionbeam" then set the flux droplists differently.
- 6. EVT:
 - Initial: If "ionbeam" then set the flux droplists differently.
 - Also on device change and template read.
 - Flux_select:
 - If "ionbeam" then set the flux droplists differently.
- 7. Version 7.3j

16 Mar, 2016

- 1. MPDA revision5: (comment w/ ;@3-16)
 - a. Da_xstep_evt2:
 - i. Made changes to parallel da_evt.
 - ii. Uses da_accumulate10 now, with no mpda_table.
 - b. Analyze_image (image_routines):
 - i. Form overlay spectra for MPDA via 'sum_region_overlay' call. See discussion in "MPDAM method data.docx", section "Revisit weighting theory5" and s/w log 14, p.143.
 - ii. The pure spectra for Back still have a charge scaling build in. These needed to be divided out (and region charge scaled in) before phase averaging of Back pure spectra to form the overlay (this is now done in 'read_da', see below).
 - iii. Do not form ayield anymore.
 - iv. Only save overlay, phase to results struct if yield map present and n_comp>1.
 - c. Sum_region_overlay:
 - i. Does per pixel, phase weighted overlay spectra per element.
 - d. Build_da_fit:
 - i. Restructure to separate MPDA, old MPDA, single phase modes.
 - ii. Use new 'multiplicity_scale' for rG sum.
 - iii. Use poverlay spectra from regions.
 - iv. New scheme solves the lack of total rG in DA matrix for Back. Now the overlay uses the image array, which has total rG divided out (in "da_evt") for all elements and Back. Then in spectrum reconstruction scaling by total rG is used for all elements and Back, which enables it to scale correctly if fewer channels are selected
 - v. Old (phase, ayield) still supported, but only if present.
 - e. Append_DA_fit:
 - i. Added penalty term for back above data.
 - ii. Added option (disabled) for higher power (than 2) for fit above data.
 - iii. Apply scale optimization for normal DA too.
 - f. Read, write regions:
 - i. Updated this to include overlay. Version -29.
 - ii. Forms valid pointers for ayield, phase, poverlay only if these terms found in file.
 - g. Scaling of Back matrix and Back:

- i. DA matrix rows need to be scaled $\text{matrix.charge}[j] / \text{mean}(\text{matrix.charge})$. Was done in 'da_evt', 'da_xstep_evt'.
 - ii. The Back pure spectra need to be scaled $\text{mean}(\text{matrix.charge}) / \text{matrix.charge}[j]$. Was done in 'analyze_image'.
 - iii. Now both are done in 'read_da' on input (only for MP DA matrices).
- h. Tested against run #111163 results for MPDA Revision 4 (see "MPDA Revised5" notes in file "MPDAM method data.docx").
 - i. Note about matrix corrected images:
 - i. The correction of images using the 'correct_yields' approach adjusts image values, pixel by pixel. This tends to upset region spectra overlays using the normal (non MP) approach, which uses the image conc values to scale pure element spectra. There is not an obvious cure to this problem, except to use MPDA, which works better.
- 2. Set_image_view (image_routines):
 - a. Called after 'image_load2'.
 - b. Re-instate notify to 'image-results-clear' to clear Regions table.
 - c. But fix error in 'clear_image_table_table' that lost pregon.
- 3. clear_image_table_table (image_table_eventcb):
 - a. Fix error that lost pregon (now set $*(\text{pstate}).p = \text{ptr_new}()$)
- 4. evt_start_image_increment2:
 - a. Test and re-normalize phase map during /flatten phase after image sums.
 - b. Test for overlapping pixels in yield map add and rescale these by 0.5 to maintain average yield.
 - i. **NOTE: This was done very badly**, which caused larger errors over many pixels. This is corrected in version 73w.
- 5. Correct_yields:
 - a. Fixed button sizes and display only name, not paths on buttons.
- 6. Region table EVT:
 - a. Added /translate, updir=3 on all 'file_requester' calls for files: linear, pileup, throttle, etc.
- 7. File_requester:
 - a. If full file not found, also apply updir to alternative 'path'.
- 8. Version 7.3i

13 Mar, 2016

- 1. MPDA revision5: (comment w/ ;@3-16)
 - a. Multiplicity_scale:
 - i. New function to form rG sum.
 - b. Da_evt:
 - i. Comment out all MPDA table construct, phase_weights
 - ii. Use new da_accumulator10 (no table or phase_weight, but w/ 1/Y)
 - iii. Use new 'multiplicity_scale' for rG sum.
 - iv. Test imaging using 111163 (Rock3) against "MPDA Revised4".
 - c. Da_accumulate10:
 - i. Return to da_accumulate6, but with per pixel inverse-yield.
 - ii. Inverse yield could be simpler than this.
 - d. Define:
 - i. Region Table:
 - 1. Add 'poverlay' pointer to overlay spectra.
- 2. Version 7.3h

8 Mar, 2016

- 1. Midas device:
 - a. Get_midas_details:
 - i. Decode XML using 'xml_to_hash' /ToStruct and assigned a few metadata from XML. Many missing still. Also fixed update header details.

- b. Xml_to_hash:
 - i. Now has default string input and “file=” keyword for files.
 - ii. Fixed recursion on ToStruct method → changed to a normal function.
 - iii. But, fails to find the new Schema URL in run00002.mid.
- 2. Maia device:
 - a. Trim_evt_files:
 - i. Extend ‘fmask’ to one blog file further to catch Y overflowing into the next blog file.
 - ii. This manifest itself as missing the last blog file sometimes.
- 3. Da_evt:
 - a. Save original phase as *phase, not the ‘phase_weight’.
- 4. Version 7.3g

4 Mar, 2016

- 1. Maia device:
 - a. Read_buffer:
 - i. For spectrum mode, set dt=0 unless “time_last” is non-zero. Otherwise, get a huge dt at start of a blog file with no header (other than “0”).
 - b. Clear borders:
 - i. Set default to zero for all axes.
- 2. Da_evt_stack:
 - a. Test for XANES stack DA as n less than 2.
 - b. Fix infinity test number of planes.

22 Feb, 2016

- 1. Blogd bug and new TAGs:
 - a. Read_Maia:
 - i. Added ‘summary4’ and ‘report’ TAGs.
 - b. Blog_read_next:
 - i. Added test on ‘AA’ and ‘BB’ validity, force a socket retry.
 - ii. Force nbytes to be less than 64K.
 - iii. Set ‘record.length’ to actual nbytes.
 - c. Blog_Browse:
 - i. Blog_browse:
 - 1. Extend tags list.
 - ii. Update_maia_records:
 - 1. Make safe for illegal tag number in both length=0 and main loops. Show unknown tag as “??”.
 - 2. Add tag 56 “summary4”.
 - iii. Update_maia_details:
 - 1. Make safe for illegal tag number in both length=0 and main loops. Show unknown tag as “??”.
 - 2. Add tag 56 “summary4”.
 - d. Blog_client_activity:
 - i. Added debug code to write to a log file, when debug=1.
 - 1. Build file-name from ‘conf_file’ path, ‘username’, ‘prefix’.
 - 2. Write debug using ‘gprint’ for each tag type.
 - ii. Write clientname AFTER again loop label, so it gets done again after a retry socket open.
 - iii. Blogd bug:
 - 1. If request ‘sendprev’ for a non-existent tag, it return a zero tag record, with len=0, but produces rubbish on next read.
 - iv. New tests and blogd bug work-around:
 - 1. Test for good ‘AA’ and ‘BB’. If bad force socket retry.
 - 2. Also test len=0; with bug force socket retry
 - a. **Later change this back to loop to ‘more’.**

- b. See comments in code.
 - 3. Never loop if tag=0; let bounce through case.
- e. Socket_retry:
 - i. Do not test and bomb after 100 retries.
- 2. Version 7.3f.

2 Feb, 2016

- 1. PIGE processing:
 - a. Analyze_image (image_routines.pro):
 - i. Test for (*p).mode = 0 before reading DA file.
 - b. Image_Process_Merge_Gamma:
 - i. Add 'Li' to list and set quant factor based on LiF sample (#1874).
 - ii.
 - iii. Set (*p).n_el to new number (9).
- 2. DAQ Control:
 - a. DAQ_launch:
 - i. Use (*pm).run.pixel for pixel coordinates.
 - b. Stage_list:
 - i. Allow it to open and Append with nothing in list.
 - c. Stage_edit:
 - i. Veto non-rectangular scans for now (Klee only has 32 raster 'steps').
 - d. Build_raster_list4:
 - i. Now have raster, fly/step, interlace and overscan modes working.
 - ii. All moves now use 'pixel' 'move.measure' in 'raster.step'.
 - iii. Sets up velocity of Stage and DACs, pixel parameters (pitch, origin, extent, enable).
 - e. Define:
 - i. Add 'run.pixel' to both DAQ *pm and shared1 structs.
 - f. DAQ client parameters:
 - i. Read 'pixel.dim[].coord' into 'run.pixel'.
 - g. DAQ_update_parameters2:
 - i. Copy 'run.pixel' from (*pdat) shared to (*pm).
- 3. Blog-Browse:
 - a. Fixed "SU" bad error, now version 7.3e.
- 4. Version 7.3e.

22 Jan, 2016

- 1. XML reading:
 - a. New routine "xml_to_hash.pro" to read XML file and convert it to an Ordered Hash (w/ option to convert this to Struct).
 - b. Uses some features from version 8 (XML 8.0, Hash 8.3).
- 2. Midas:
 - a. The xml_to_hash routine works for Midas file 412, only if the Schema entries are removed (i.e. all parameters on <odb> line).

11 Dec, 2015

- 1. Midas
 - a. New device driver for format version 4. The format has short LONG for Q and Dwell in new LONG STAT record, and a new record DTME with entries for DT Corr factors for each detector. These are not used yet.

9 Dec, 2015

- 1. Unstringify(): fails if string contained unbalanced container characters.
 - a. Hide_embedded:
 - i. Should ignore all container characters within a text string and not change level 'lev'.

7 Dec Oct, 2015

1. FalconX
 - a. New device driver
2. Midas
 - a. New device driver for format version 3. The format still has short INT for Q and Dwell, and no entry for DT yet. Will need another revision.
 - b. Bug in format (some ADCx sizes are short by 4) sometimes crashes the Fortran.

2 Oct, 2015

1. Sync GeoPIXE-source to GeoPIXE-source2
 - a. Version 7.3c

9 Nov, 2015

1. image_lib.f:
 - a. maia_accumulate_dtfx (and 3D):
 - i. For old blog data w/ Epics PVs in Monitor events, and no extra words for BT or Flux1 or Flux2, these may not have any 'veto' pseudo events at all. But all normal events (w/ x,y,E) have the epics PV jammed in their maia_fx[0,*] array member. It seems that the new 'maia_accumulate_dtfx' (and 3D version) ignores the veto=0 events. Corrected this error.
2. image_correct_flux:
 - a. Only set 'use_FC_index' to a value within range of 'n_attributes' available, else set it to -1 to use default flux[*,*,0].
 - b. This assumes that the added attributes will always be first in list after call to method 'check_pv_list' in device.
 - c. Added notes in Maia and base device to emphasize that added attributes must be first after method 'check_pv_list' in device.
3. Da_evt_tomo:
 - b. Also write out the flux 3D array as a "Flux" stack.

21 Oct, 2015

1. standards wizard:
 - a. The enable is set to off for an incomplete row entry.
 - b. Two plots now, one versus run #, one versus line energy.
 - c. Has an "old" and "new" detector droplist to test effects of changing detector calibration – only affects plots on "plots" tab, not the table.
2. Image regions:
 - a. Export of long lists (e.g. XANES energy stack regions) will only display checkboxes for leading and trailing 50 columns.

12 Oct, 2015

1. Missing rows in Cluster sort image
 - a. When using ycompress, there are partial rows after reconstruction of full image.
 - b. Need to compress before subtracting max-min+1 to for range.
 - i. Re-order instructions to get correct offset3 and yrange3 values.
 - ii. Done correctly in Maia device already.
 - c. Corrected in: da_evt, cut_evt, da_evt_stack, da_evt_tomo.
2. Maia Control, sub-processes
 - a. Fixed background process, not shutting down, issue:
 - b. 'spawn_blog_client' and 'spawn_maia_client' both use /noshell for Linux.
3. Parallel_client:
 - a. Use same approach for cluster processes on Linux.
4. Region 'spec_evt' processing in cluster mode:

- a. Pass linear, pileup, throttle, even in mask mode to 'spec_evt' to override the values in the region array.
 - b. Return to using 'mfile' for all (other) parameters in regions.
5. Wizard_standards: scan dir:
 - a. If 'sample' is not found, but 'serial' and 'detector' and 'energy' are, then copy back the config 'name' as new 'sample' and update this in table.
6. calc_slices2:
 - a. Fixed error in 'ns' line #104, which gave "Array subscript for SLICES must have same size as source expression" error.
7. evt_start_image_increment2
 - a. Image increment calculated dy 1 pixel short; fixed.
8. Version 7.3c.

8 Oct, 2015

1. Region file-names issue in Cluster mode
 - a. Fixes (in Image_Table_EVT) to pileup, throttle, linear paths not passed if using /cluster and 'mfile' passing.
 - b. Have reverted to stringify whole mask array. But this will limit both the number of regions and the total number of pixels selected (in 'q' pointers) as the parameters 'args' shared memory is only about 10 Mbyte.
2. Standards wizard problem:
 - a. Region results became undefined. It was cleared in Table when it cleared the table after the sort. This could be delayed to after the new region was analyzed.
 - b. Removed the table clear from Table notify routine, as it is done already in Image as part of 'set_image_view'.
 - c. Version 7.3b.

2 Oct, 2015

2. Synced GeoPIXE-source2 to GeoPIXE-source
 - a. Included fix to mpdam, flat flux, spec_evt Interpol, spawn issues.
 - b. Version 7.3a.

30 Sep, 2015

1. Maia Control
 - a. close_file:
 - i. Try to use /force (passed as _extra) to force kill of spawn'ed processes.
 - b. spawn issues solved:
 - i. 'spawn' by default forks a shell process that runs the command. The PID returned is for the shell fork and not for the command. This seems to frustrate the use of the unit close to kill the process.
 - ii. If we use '/noshell', then the PID returned is correct and the unit close mechanism seems to work. But then we need to pass the command and args as a string array (and with no "" bracketing command sav file).
 - c. spawn_blog_clients:
 - i. Changed spawn to use /noshell and a string array for arguments.
 - ii. Seems to handle the 'conf' file with spaces in name OK this way.
 - d. test_spawn_close:
 - i. Test routine to try above scheme, uses sub-process 'blog_client_test'.
2. spec_evt:
 - a. The flinear dither scheme is wrong. replaced it with a call to 'interpol' to do a linear interpolation on 'e+R' into 'flinear' array.

5 Sep, 2015

1. MPDA method2
 - a. Build_da_fit:

- i. The component phase Back pure spectra implicitly include multiplicity as they were fitted with all detectors. Hence, don't need full rG scaling here.
 - ii. However, take care if # detectors changes between building DA matrices (and in 'matrix' struct) and applying them to region spectra here.
- b. image_routines: analyze_image
 - i. Error in ayield[i] code: $x2 + y2*\underline{ysize} + \text{pixeln}2*I \rightarrow x2 + y2*\underline{xsize} + \text{pixeln}2*i$
- c. da_evt:
 - i. Use 'finite_image()' to make sure yield maps are finite, replacing bad pixels with average. Use this for 'image' and 'image_error' too.

4 Sep, 2015

1. Copied GeoPIXE-source2 to GeoPIXE-source
 - a. Followed instructions in file "Move GeoPIXE2 (develop) to GeoPIXE (stable).docx".
 - b. Corrected script and pro files, and also Project Compile Properties in IDL, as instructed.

2 Sep, 2015

1. Wizard Import Fit: import a series of spectra and fit each
 - a. Select the device and format type using 'import_select'.
 - b. Scans a selected dir for raw data files and extract metadata.
 - c. Import each, delete XYT, Cal All, sum-spectra, fit and save.
 - d. Use new approach at end of each call-back routine to test and increment loop counter. Note: Loop index is not the same as table index (row).
2. spectrum display OnNotify:
 - a. 'import-spectra': Pass struct of parameters from Wizard_import_fit, return any changed file-names and charge.
 - b. New Wizard commands for 'import-spectra', 'sum-spectra', 'energy-cal' and 'save-spectra'. Old 'sum-spectra' renamed to 'sum-selected-spectra'.
3. spectrum_load_do:
 - a. Pass extra pars to accommodate call from Notify from 'Wizard_import_fit' (charge_mode, flux_pv, conv, charge, verify, sret, err).
 - b. If any of these are passed (and not zero), set the (*pstate).pflux struct value.
 - c. 'charge' will be updated if set during import, else this value is used in spectra.
 - d. If /verify, check all file-names first (/skip_if_exists) and return any changed file-names (like in 'evt_start' for images). Still need these file requesters in 'spectrum_load_prep' in case one is needed but blank initially (e.g. old data).
4. spectrum select:
 - a. New Wizard commands 'delete' to delete by various selection methods.
5. fit setup:
 - a. Found that a bad fit could set w0, w1 to crazy values in detector. Disabled 'fit_memory_on' = 0 (in 'image' and various 'c_fit_memory_1' tests in several pixe_fit routines).
 - b. New Wizard command to 'fit-one'.
6. flux_select:
 - a. Can pass parameters to set-up pflux struct and return /silent without pop-up.
7. Depth wizard:
 - a. Now uses 'sum-selected-spectra' command to Spectrum Display.

31 Aug, 2015

1. file_requester: bad dir open/create errors
 - a. Behaviour of 'safe_file_mkdir' reports dir not writeable, when it is.
2. safe_file_mkdir:
 - a. If 'file_test()' reports dir not writeable, try a test write.
3. write_test:
 - a. New routine to try to write a file, and delete it afterwards.
4. find_file2:

- a. Use “/expand_tilde” on calls to ‘file_search()’ for “~” to work under Windows. This is the default for Linux anyway.
 - b. Called from ‘grow_tree’.
5. Version 7.2c

14 Aug, 2015

1. Background processes in GeoPIXE: They don’t seem to stop and close sometimes. On Linux, the close port approach does not always work. Need to force cancel in background process too.
 - a. ‘cluster_client’ uses ‘parallel_release’ after a ‘cancel’
 - b. parallel_client:
 - i. parallel_release:
 1. Sets (*ppar)[3]=1 if ‘cancel’ now.
 - c. ‘da_evt’ calls ‘worker_progress’ to test for ‘cancel’, but did not get it.
 - d. parallel_worker:
 - i. worker_progress:
 1. Needs to test if foreground process (cluster_client) has set a kill condition after a ‘cancel’, which is done via ‘parallel_release’.
 2. if (*ppar)[3] equals 1 then return,0
2. Background processes in Maia_Control: They all should respond to (*ppar)[3]=1 and shut-down. Need to give them time to do this. So now:
 - a. Maia_launch cleanup:
 - i. sends (*ppar)[3]=1 to all first, closes ports too
 - ii. then sets /hourglass and waits 10 s
 - iii. then destroys widgets.
3. Background processes in Maia_Control: They don’t seem to honour the close port always. Now do that first and wait 10s to be a bit safer (see above).
 - a. We depend on the sub-process seeing (*ppar)[3]=1. But some wait a long time for a read from blog (e.g. Epics perhaps).
 - b. But most have a 3s timeout on read set, and test for (*ppar)[3]=1 on read fail, so they should not go more than about 3s before finishing.
4. Version 7.2a (7.2b for minor fixes during workshop).

6 Aug, 2015

1. Maia_Control:
 - a. Changes to PV val and unit, when not using H/W scalars, did not set values right. Fixed.
2. Shared memory:
 - a. parallel_client:
 - i. Use ‘get_login_info()’ to get username to build a more unique shmmap ‘prefix’ for shared memory (e.g. “ryal13_geo_1_”).
 - b. geopixe_parallel:
 - i. Use ‘get_login_info()’ to get username to build shmmap ‘prefix’.
 - c. shared_memory_buffers:
 - i. Add the ‘destroy’ keyword to pass to SHMMAP.
 - ii. Use this in ‘parallel_client’ parallel_config routine.
3. Version 7.2

29 July, 2015

1. Header read Maia:
 - a. Needed to extend header read to 7 MB to catch all var-val metadata.
2. Maia_Control:
 - a. Change version test for ‘flux.chan’ data from Kandinsky to 6646 (not 6664 as set at DESY) – in Maia-client-parameters-slow, Maia_launch_read_da_info.
 - b. Also, must never copy the (*pdat).IC.pv.name to (*pm).IC.pv.name. The former has the XFM name, while the latter will retain the ‘Maia:scaler.FC0’ name if in h/w mode. Changed in ‘maia_launch_read_da_info’ and ‘maia_setup_update_dynamic’.

3. Version 7.1_

24 July, 2015

1. read/write image files:
 - a. Move Device Object write earlier (version -55) to just after preview, so we can skip all plane reads for a /header read.
 - b. New version -55.
2. image_details:
 - a. This means that we do not have flux map for 'image_details', so we estimate flux from charge/conv instead.
 - b. Veto doing stats for images over 10M pixels. Add a new /stats keyword.
3. Image menu:
 - a. Added a new menu item for "Properties & Pixel Statistics" to avoid calculating pixel statistics is not needed.
4. spectrum_load_new:
 - a. Revamp this in preparation for a Batch Importer ...
 - b. Do all file requester stuff here.
 - c. Call 'flux_select' here (but will need to update of 'sensitivity' from file header read and set 'preamp.val' and 'preamp.unit' later).
 - d. Pass parameters: F, F2, throttle, pileup, linearize to 'spectrum_load_do'
 - i. pflux, path are part of *pstate struct.
5. spectrum_load_do:
 - a. event: event.top → group, pstate.
 - b. update of 'sensitivity' from file header read to set 'preamp.val' and 'preamp.unit' before file import.
 - c. do the import → p

10 July, 2015

2. FalconX device:
 - a. Falconx_device__define:
 - i. Based on Maia to start with.
 1. Keep ADC re-direction, clear borders, XYZ axes, slow axis.
 2. Keep YLUT, still assumes Y grows from zero though.
 3. Do not use Flux for now (needs flux counters later), but make sure the flatten code is done.
 4. assumes "*_nnnn.dat" raw data files.
 - ii. Read_buffer:
 1. Pass X1,Y1,Z1 as INT to allow negative coordinates to pass back.
 2. Convert to UINT after offset done for return.
 3. Simply weighted DT, no Flux.
 - b. falcon_events1: Fortran:
 - i. Pass X1,Y1,Z1 as INT to allow negative coordinates to pass back.
3. DA_evt:
 - a. (allow negative offsets in "Windowing")
4. EVT_start:
 - a. do not clip XY offsets at zero (allow negative offsets in "Windowing")
 - b. Pass Z compress to da_evt_tomo
5. spectrum display: Spectrum_Add
 - a. Allow spectra up to 16k for Add.
6. da_evt_tomo:
 - a. Pass Zcompress from evt via evt_start.
 - b. Do not pass el_select or energies file.
 - c. Use 4D image array, but save as a series of 3D XAN files w/ unique names.
 - d. Note different image order of 'x,y,el,z' versus XANES of 'x,y,z'.
7. da_accumulate_tomo:

- a. Like accumulate5, but with Z axis on image, error.
- 8. Sort EVT:
 - a. Add scan mode 5 (XYZ tomo).
 - b. Add Z compress, visible in scan mode 5 only.
 - c. "Cluster" will only work if Slow Axis = Y axis. If not an error pops up.
- 9. Define(/stack) add:
 - a. stack_type: 0=XANES, 1=Tomo
 - b. zcompress
- 10. read/ write geopixe images:
 - a. Z compress, stack_type
- 11. image_correct_flux_stack, image_flux_flatten_stack:
 - a. Note different image order of 'x,y,el,z' versus XANES of 'x,y,z'.
- 12. cal window, line_energy:
 - a. Assumes "Ge" detector when called from Cal window.
 - b. Change to "Si" default.
 - c. Allow specific escape codes: "Si-esc", "Ge-esc", "Ge-esc1", "Ge-esc2" in addition to defaults.
 - d. Add these specific codes to Cal droplists.
- 13. Version 7.1~

3 July, 2015

- 1. Bugs in XANES stack format read/write:
 - a. To handle new 3D XANES, we added 3D flux array write with version -50 of XAN file. However, code for building stacks still saves only a 1D flux. So to handle 1D flux in XANES stacks added write of n_flux in version -52.
 - b. write_geopixe_image:
 - i. For XANES mode, after has_flux, write n_flux (# planes for flux).
 - c. read_geopixe_image:
 - i. For XANES and version le -52
 - 1. Read n_flux and then loop on n_flux for planes
 - ii. Else for XANES and version le -50
 - 1. Assume n_flux equals # element planes.
 - iii. Else for non-XANES or (XANES and version gt -50)
 - 1. Assume n_flux equals one plane only.
- 2. New version 7.1>

13 June, 2015

- 1. Bugs in Traverse/ProjectX:
 - a. Undefined: ayield
 - i. ayield in results used array bounds arguments → bad if undefined.
 - ii. Pass ayield as undefined OK into ptr_new() if do not use array bounds.
 - b. Bad DevObj in region #0 after #1 created.
 - i. Needed a separate clone of spectrum DevObj when forming results as well as for each traverse spectrum.
 - c. Corr spline, PCA spline:
 - i. Fixed display of spline during move. Needed to not clear all in Notify of 'clear-analyze', just clear spline.
- 2. New XYZ sorting directly into a XANES stack (see "NMP Petra May-2015 notes"):
 - a. maia_384_events6:
 - i. New Fortran to pass Z0 and return Z vector.
 - b. Maia_device:
 - i. Use maia_384_events6 to acquire x,y,z on axes 0,1,2.
 - ii. Need to add redirection to map between these.
 - iii. Calls 'maia_accumulate_dtfx_3D' for the x,y,z case to build flux, dead_fraction and dwell.

- iv. Later need to remove /image_mode from this Fortran.
- v. Add an encoder mode for axes 1,2.
- vi. Add XYZ clear widgets, to use with encoder correction modes.
- vii. New parameters saved in version 3,4. Always add new items at end so we can always read device parameters, even from an older driver.
- viii. Read_setup: Uses 'save_header_data' and 'update_header_info' methods to update header info. However, do not depend on it (e.g. if blog .0 file is not used).
- ix. Get_header_info: now uses new 'save_header_data' and 'update_header_info' methods after reading file.
- x. New sort options widgets:
 - 1. Device re-direction XYZ
 - 2. Clear X, Y, Z
 - 3. New encoder mode
 - 4. x_margin not used anymore.
 - 5. slow axis
- xi. Added to self:
 - 1. sort_options:
 - a. source.x,y,z: source axis index (0,1,2) for redirection.
 - b. clear,x,y,z: clear border values
 - c. slow_axis: for more general YLUT (must equal Y axis selection for now)
- c. Base device:
 - i. register_notify: register a notify to parent TLB.
 - ii. update_notify: send any registered Notify, called in Maia device when device re-direction controls changed.
 - iii. save_header_data: store away raw header data
 - iv. update_header_info: use raw header data to fill in self.header.
 - v. show_header: returns the header struct.
 - vi. set_header: uses 'copy_pointer_data' to populate a header. Misguided really – needs a full 'get_header_info' to be of use.
 - vii. Added to default self:
 - 1. Notify details: on, tlb, from, message
 - viii. Added to header:
 - 1. sensitivity, IC_name
 - 2. Z pixels, origin, size, XYZ: on, name
 - 3. XYZ coords and units
 - 4. old_mp: to store raw header data
 - 5. default detector = -1 for all channels.
- d. Generic device_specific, read_buffer:
 - i. New fields in header for Z origin, mm, pixels, pz_coords, x_name, x_on, etc. all passed only via _extra mechanism.
- e. Slow axis needed for YLUT:
 - i. in Read_buffer: set maia_y_min to whatever is OK. The problem is that YLUT also linked to using Yoffset and Yrange3 in da_evt, etc., which means that we can ONLY have Y as the slow axis for now.
 - ii. This means for cluster processing we can only have full 3D modes such as: EXY, XEY, theta-XY, X-theta-Y (theta-YX will not work).
- f. da_evt_stack:
 - i. Build the stack directly for 'el_select'.
 - 1. Uses new Inputs:
 - a. zrange, scanz, zorigin
 - b. translate_energy table (zrange long)
 - c. el_select
 - 2. Pass in as table 'translate_energy', which should match Z pixels for size. Where does this come from?

- 3. Uses a `e_lookup` table to access the correct DA matrix rows.
 - ii. Uses new 'stack' forms of the flux correction routines.
 - iii. Only corrected dead-time and pileup per XY pixel so far.
- g. EVT:
 - i. Added "XANES Stack" mode=2.
 - ii. Add `xy_scan` mode=4 for "Scan X,Y and Z=E".
 - iii. Add Z widgets on Scan tab; map on when in XYZ mode.
 - iv. Added: XANES element, XANES energies file.
 - v. Detector Notify from Maia device and 'evt_check_mp' to update flow-on widgets (e.g. sizes, origin widget updates).
 - vi. Use the new 'register_notify' method, apply to ALL devices in list.
- h. `evt_start`:
 - i. In cluster mode, do all with `flatten=0`, then flatten afterwards in increment routine.
- i. `evt_start_increment2`:
 - i. Deal with XANES case properly. Call 'stack' variant routines.
- j. `write_geopixe_image`:
 - i. xanes 3D maps, save 3D flux array.
 - ii. Allow various arrays in stack mode (controlled by `has_*` flags):
 - 1. dead, dwell, pileup, rates, phase, yields
 - iii. Version -50
- k. `read_geopixe_image`:
 - i. Deal with new arrays for (xanes eq 0) or (version le -50).
 - ii. Version -50
- 3. Blog-browse:
 - a. Show Z PA on summary list, as well as Z dimensions for scan info.
- 4. `copy_pointer_data`:
 - a. Match tags between structs, so the tags do not have to be in the same order.
 - b. Clone object on /init first for object type `psrc`.
 - c. Destroy old objects after cloned new one.
 - d. Now use 'copy_object_data' for copies.
- 5. GeoPIXE and Maia-control version 7.1+ → 7.1=, 7.1<

28 May, 2015

- 1. New charge values and units handling:
 - a. `charge_sensitivity()`:
 - i. to convert input 'val', 'unit' to 'sensitivity' in "nA/V"
 - ii. will accept "nA/Hz" form too, and is NOT case sensitive
 - b. `find_charge_val_unit_index()`:
 - i. to convert 'sensitivity' to index for val and unit arrays/droplists
 - c. `charge_val_unit_lists`:
 - i. to return standard lists for vals and units for droplists
- 2. Routines updated to use new charge routines:
 - a. EVT:
 - i. template from, val, units droplists, scan PVs, `evt_set_preamp`, `evt_set_preamp_units`, `onrealize val/units`, set-up widgets.
 - b. `flux_select`:
 - i. as used in spectrum Import.
 - ii. why is this not "generic_flux_select"?
 - c. `generic_flux_select`:
 - i. as in Maia device :: `flux_scan`
 - d. `get_maia_details`
 - i. as in Maia device and blog-browse.
 - e. `image_details`, `spectrum_details`
 - f. `maia_setup`:
 - i. val, units droplists, scan PVs, `maia_setup_update_dynamic`

3. `maia_control`: use `flux.chan[]` and distinguish between IC, IC0, IC1, ICE
 - a. `maia_client_parameters`:
 - i. reads DA info from scratch and puts in shrmem
 - b. `maia_client_parameters_slow`:
 - i. read `'flux.chan[...]` variables and set shrmem `*pdat.IC0, IC1`.
 - ii. New IC struct item "remote" indicates that a flux unit was found in `'flux.chan'` variable and hence flux sensitivity is controlled by beamline, and should not be settable from Maia-Setup UI.
 - c. `maia_launch_read_da_info`:
 - i. read from shrmem `*pdat` and set `*pm` for IC0, IC1
 - ii. decode DA info string from scratch into ICE.
 - iii. update IC from IC0, IC1 or ICE depending on PV selected.
 - iv. But don't copy the `pv.name` to `IC.pv`, when the PV is changed in droplist, just the usual "Maia:scaler.FC0", ...
 - d. `maia_update_parameters`:
 - i. Calls `'maia_launch_read_enable'`, which calls `'maia_launch_read_da_info'`.
 - ii. Notify 'maia-display' to Maia-setup to refresh widgets.
 - e. `maia_setup`:
 - i. widgets updated when tab changed: `'maia_setup_update_dynamic'`.
 1. 'maia-display' notify: do not refresh flux widgets here.
 - ii. writing of a new DA info into scratch in `'maia_setup_apply_imaging'`:
 1. Image DA set-up changes do this, only important for ICE ones as these are the only ones used on read-back if IC0, IC1 remote.
 - iii. Use `('*pm).IC.remote'` to veto changing val, unit droplists.
4. Deadtime parameters: New ability to set them "Auto":
 - a. `maia_client_parameters`:
 - i. Copy `scratch.datum[DT]` into shared memory as a byte array.
 - b. `maia_launch_read_info`:
 - i. Decode byte array from shared memory (or read Kandinski) to set Auto.
 - c. `maia_update_parameters`:
 - i. Calls `'maia_launch_update_deadtime'` to set DTcalA is in auto mode.
 - d. `maia_launch_update_deadtime`:
 - i. If deadtime in auto mode, set DTcalA from a table based on TDS and Scepter version.
 - e. `maia_setup`:
 - i. New "Auto" checkbox for Deadtime parameters.
 - f. `maia_setup_apply_hymod`:
 - i. Sends the new "{auto:1}" string to scratch.
 - g. `maia_setup:update_dynamic`:
 - i. Update Auto widget.
5. Maia header:
 - a. `get_maia_details`:
 - i. Add "IC_name". Set this to "Maia:scaler.FC0" if flux0 explicitly used.
 - ii. Extend Petra jkluge for scan.info (in `var_val`) to include: IC_sensitivity0 (then set IC_name too), grain, comment.
 - b. `base_device__define`:
 - i. Extend header to include IC_name.
 - c. Maia device:
 - i. Pass on IC_name, sample, title, grain in header.
 - d. `evt_check_mp`:
 - i. If array, set array mode and all Cals.
 - ii. If sensitivity found, set preamp sensitivity and units droplists and make sure in charge mode 1 (w/ PV).
 - iii. If IC_name found, set the PV droplist too.
 - e. `spectrum_load_new`:
 - i. Also pass on sample, grain and comment from header.

18 May, 2015

1. Base device:
 - a. Add to header:
 - i. Pileup and throttle now have “found” to indicate that some genuine (non null) info was found in the various header details, as distinct from “on”, which means enabled in Maia.
2. Get_maia_details:
 - a. Set “found” if a valid record found (var_val, scan.info, metadata).
3. Maia Device:
 - a. Use new pileup, throttle “found” from ‘get_maia_details’
4. Spectrum_display_new (from Import):
 - a. Use “found=0” or “on” to open file-requester(/skip_if_exists).
5. Spec_evt, spec_xstep_evt:
 - a. Added throttle file info to output file.
6. file_requester():
 - a. pop-up errors if fail to safe_make_dir on path text edit.
7. safe_make_dir:
 - a. Add /verbose switch, used with path text edit.
 - b. Pop-up error if not dir, or write protected.
8. Version: 7.1z

15 May, 2015

1. Maia device:
 - a. Refine reading of header to:
 - i. Include new header entries (see ‘get_maia_details’).
 - ii. Avoid re-reading header for same file-name (e.g. flux scan).
 - iii. Includes ‘var_val’ scratch variables for pileup and throttle.
 - b. Read_maia_32_header :
 - i. If file is the same as last (stored in common), then use previous header.
 - ii. This does mean we need to do a good search the first time. For this reason, we avoid using the /no_scan option elsewhere.
 - iii. Use 2 MB read by default (no 0.5 MB).
 - iv. Only read a single *.0 blog file; re-read it for longer search.
 - c. Get_maia_details:
 - i. Expand var_val read for cal, deadtime parameters, Hermes ECH, pileup (old .info), throttle (old .info) and the scratch.datum values. For scratch, pick out the pileup and throttle.
 - d. Read_maia:
 - i. Now reads 64k bytes always to get long var_val records.
 - ii. Will make initial Blog-browse read slower (and use more memory?).
 - iii. Change ‘record.b’ to be a pointer, only as large as is needed (will need to be freed later).
 - iv. Use record.b as a pointer instead of fixed length string.
 - e. Free_record:
 - i. Use new ‘free_record’ to free record pointer struct arrays in ‘build_maia_ylut’, ‘read_maia_32_header’ and ‘blog_browse’.
 - ii. Also in error catching in ‘read_maia_segments’, ‘blog_browse’, ‘build_maia_ylut’ and ‘read_maia_32_header’.
 - f. Get_header_info:
 - i. Includes new items in header: cal.
 - ii. Does not use /silent, which uses /no_scan in ‘read_maia_32_header’.
2. EVT:
 - a. evt_check_mp:

- i. Use new “evt_set_...” calls to set parameters from header.
 - ii. Add: pileup, throttle.
 - b. Start button:
 - i. Also use the “evt_set_...” calls after checking files. Only set if file name changes.
- 3. Wizard_standards:
 - a. New wizard draft.
 - i. Uses new header details and metadata if present.
 - b. Evt:
 - i. /verify on ‘evt_start’
 - 1. Now checks files in ‘evt_start’ and returns old/new for any changes.
- 4. Table edit WIDGET_TABLE_CH under Linux:
 - a. Detect both <nl> (10B) and <cr> (13B) in ‘WIDGET_TABLE_CH’ events to indicate a carriage return entry.
 - b. Add to all editable tables: Spectrum select, image regions, batch sort, wizard_standards.
- 5. Version 7.1x

7 May, 2015

- 1. Image_lib.f
 - a. Added ‘mdaq_events’. Like ‘mpsys_events’, but with no offset or X stepping.
 - b. Nearly lost new changes – got image_lib.f back from backup.
- 2. MicroDAQ device:
 - a. New ‘get_mdaq’ to read the .mdaq file, similar to .mp file.
 - b. Like MPsys device, but with no 2K XY offset, and new .mdaq parameters
 - c. GeoPIXE version 7.1v
- 3. Wizard_standards:
 - a. Evt:
 - i. ‘sort-image’:
 - 1. Accepts struct of Sort EVT window parameters.
 - 2. More general than old ‘sort-with-details’.
 - b. Image:
 - i. ‘sum-region’:
 - 1. Makes a Box shape of ~95% of the area and hits analyze.
 - 2. The region results pointer is returned in wizard pdata struct
 - 3. Callback routine calculates ‘conv’ and adds to Table.
- 4. Wizard_depth:
 - a. Changed ‘Sort EVT’ wizard_notify to use more general command ‘sort-image’ instead of ‘sort-with-cal-select’.
 - b. Tidied up other call parameters.
 - c. GeoPIXE 7.1v needed.

22 Apr, 2015

- 1. Write_geopixe_image:
 - a. In /dump mode, also write out variance image size and var images
 - b. Keep version = -48, which invalidates old .bin files.

13 Apr, 2015

- 1. Results properties:
 - a. Apply for STIM correction upset lack of ‘array’ specific code there.
 - i. Added lines to read in ‘payout’ from ‘(*detector).layout’ file before case statement on mode..
 - b. 0:
 - i. Added extra parameters to ‘geo_array_yield’ calls (2) and following ‘make_peaks’ call for ‘array’, ‘detector’, ‘payout’, ‘mu_zero’, ‘ratio_yield’, ‘ratio_intensity’.
 - c. 1:
 - i. No array code done for fluid inclusions yet (needed).

- d. All:
 - i. Now all errors 'goto, finish' where all temp pointers freed.

24 Mar, 2015

1. Large 'pmask' to 'spec_evt' in 'cluster' mode:
 - a. Region mask can be too large for passing to 'cluster' processes. Could be as large as 20 (regions) * 20M pixels each * 10 (string encoding) = 4 GB (current buffer only 10 MB), to be passed to each process (replicated in shared memory) * 12 processes!
 - b. Image_table_EVT:
 - i. In cluster mode pass 'mask_file' instead of pmask.
 - ii. Must ensure that this file-name is current.
 - iii. Test results for not blank or "null".
 - iv. NOTE: Will force 'wizard_depth' to save regions before each extract.
 - c. Spec_evt:
 - i. If no 'pmask' passed, look for 'mask_file' and read_regions().
 - ii. Remember to free_regions at end, if a read has been done.
 - d. Parallel_client:
 - i. Parallel_config:
 1. Save defined buffer size in common 3, c_parallel_bufsize
 - ii. Parallel_start:
 1. Test size of args against buffer size and clip and post error.
 2. Make sure that passed byte string is blank and (*ppar)[2]=0 passed to start processing.
 - e. Geopixe_parallel:
 - i. Test length of passed arg string. If blank, cleanup and exit.
 - f. Evt_start:
 - i. Test results for not blank or "null".
 - g. ImageRGB:
 - i. Change to high now will Notify back on Image window.
 - h. Geopixe_environment:
 - i. Changed Temp dir to return user ~/temp dir instead to avoid write issues on shared Linux systems for /tmp/geopixe.
 - i. New version GeoPIXE 7.1s.

13 Mar, 2015

2. Maia_launch:
 - a. Fixed false clearing of the scratch.datum[].values even if .info fields not used.
 - b. New version Maia_Launch 7.1o
3. Maia_setup:
 - a. After HYMOD gain-trim adjustments, the save button now updates Gain Trim2 file-name.
 - b. The Arrows, Apply and Save are vetoed if gain-trim is not enabled.
 - c. New version Maia_Launch 7.1q
4. GeoPIXE:
 - a. Corr:
 - i. Fixed export of PNG – need to use /bare on pixel conversion to exclude new axes.
 - ii. Fixed export of WMF, which was going to CGM always.
 - b. Spectrum_display:
 - i. Fixed WMF export, which would clear the 'previous' output file. Now changed file for device to 'null' before switching back to Xwin. This fixes the problem.
 - ii. Put same fix in plot_corr and plot_spectra too.
 - c. Fit_setup:
 - i. Test for Q=0 to veto generation of DA matrix for zero charge.
 - ii. Add a new iThemba DMM binary export to the "Export" button.
 - iii. Still need to fix the content of the pop-up panel.
 - d. New version GeoPIXE 7.1p

- e. EVT:
 - i. Last file did not follow a changed first file (after new translation table, etc. stuff). Added a test for a change of 'first' path relative to 'final' to use 'first' path and clear file name default.
- f. New version GeoPIXE 7.1q

16 Feb, 2015

- 5. Correct_yields:
 - a. Enforce constraints on negative mineral components in both 'correct' and 'project'.
 - b. For pixels with component sum>1, normalize all components down.
 - c. However, correction retains negative pixel concs.
 - d. To project minerals, do median(3) first to each.
- 6. Image_dll.f90, image_lib.f:
 - a. da_build_mpda_table:
 - i. Remove constraint on table entries > 0. Zero ones will not get used.
 - ii. Make test against small not 100*small to collect smaller peaks (e.g. Ba Kb2).
 - b. Da_accumulate8:
 - i. Increment 'fny' with Table[] too, to match 'invy'.
 - ii. Correct error in 'multiple' branch for 'fny'.
- 7. Da_xstep_evt2, da_evt:
 - a. Add MPDA code like in da_evt. Point 'evt_start' to this new routine.
 - b. Zero 'mat_inv_yield' for any too small 'matrix.yield'.
 - c. Form 'yield' from 'fny/invy' simply now. Any NaN/Inf will be caught elsewhere.
- 8. Read_DA:
 - a. Return charge as vector across all phase fits in MPDA mode.
- 9. Build_da_fit:
 - a. When forming phase weighted pure spectrum, for "Back" scale by 'charge/matrix.charge[j]' for phase 'j'.
- 10. Build_da_phase_weights:
 - a. Add test for 32 bit processor/IDL and use smaller dimensions for Table.
- 11. Image_routines/ analyze_image:
 - a. New code for matching indices between 'image' and compressed 'yield' maps to form 'ayield' averages.
- 12. New version "7.1n".

9 Feb, 2015

- 13. File_requester:
 - a. Revised translation table code. Fixed error with new path.
- 14. EVT:
 - a. Simplify calls to file-requester, as it does more internally.
 - b. Start code now checks all paths prior to calling EVT_start.
- 15. Analysis_region:
 - a. Need to skip use of 'pda', 'phase', 'yield', 'has_yield', 'ayield', etc. which is only used for new MPDA code.

3 Feb, 2015

- 1. Maia device:
 - a. New records in blog files can be longer than 32k, so the 'skip' int variable passed to/from the Fortran 'maia_384_events4' routine can become apparently negative.
 - b. Now define 'skip' as long and call 'maia_384_events5' instead.
- 2. maia_384_events5:
 - a. Now uses long 'skip' and calls Fortran 'maia_384_events5' instead, which uses long 'skip'.
- 3. Info to scratch:
 - a. Start moving to use the new 'scratch' variables in Kandinski instead of:

- i. 'cal.info', 'DA.info', 'gaintrim.info', 'group.info', 'linearise.info', 'pileup.info', 'throttle.info', 'deadtime.info', 'ROI.info'
 - 1. Read in 'maia_client_parameters', various 'maia_launch_' read/info routines.
 - 2. Written in 'maia_setup'
 - ii. 'scan.info' – used for comment columns, not used in Maia-Control (?)
 - b. New 'scratch.datum' variables:
 - i. Cal.info → key = "Maia:cal.info"
 - ii. DA.info → key = "Maia:da.info", but not sensitivity?
 - c. Find_scratch():
 - i. No. Will instead use a fixed range of index from 11-19.
 - d. Scratch index assignment:
 - i. Need to assign scratch register indices (11-19) for purposes.
 - ii. See file "device/Maia/maia_scratch.def"
4. New version "7.1k".

24 Nov, 2014

1. Read_fit_results:
 - a. Cal this from 'load_fit_results' in 'fit_results' now.
2. Peak_stats spectrum plugin:
 - a. Determines the standard deviation (in channels) within VIEW markers for all detector channels/spectra, and forms a map across the detector array.
 - b. Calls 'read_fit_results'.
 - c. Assumes Maia 384C layout for now.
 - i. Perhaps need a pop-up field to specify the layout file?
3. Peak_fit_stats spectrum plugin:
 - a. Reads a fit-results PFR file and plots FWHM across the detector array.
 - b. Uses a pop-up to input PFR file-name.
 - i. Perhaps need a pop-up field to specify the layout file too?
4. Maia-launch:
 - a. Split off detector parameters on main window.
 - b. Added "ddm.rewrite" to the "Reset" button code to re-write DAM settings.
 - c. Make sure first and last of 'colweight' array are zero to handle clipping of 'column' in the FPGA.
5. Socket_command_get:
 - a. Sometimes get partial reads, so the next read fails with incorrect 'token' (which is actually the rest of the previous command return).
 - b. Solution on 'bad_read3' error is to read another line into 'sret' to clear buffer.
6. Plugins:
 - a. Peak Mn stats:
 - i. Forms standard deviation of peak within VIEW.
 - ii. General purpose template for later.
 - b. Peak fit FWHM:
 - i. Reads a fit-results PFR file and plots FWHM map across detector.
 - c. Peak Mn Centre:
 - i. Uses centroid within VIEW (assumed to be Mn K lines) and positions these in energy.
 - ii. Assumes a gain of 0.0088 if spectrum is not calibrated already.
 - iii. Simply adjusts offset, keeping gain fixed to centre peaks.
 - iv. Use this as a prelude to fitting ⁵⁵Fe spectra.
7. Fit-setup:
 - a. Added a new "Cal" tab with "Free Gain" control to force Gain E Cal fitting off if there are too few peaks. Sometimes the automatic approach does not work as it thinks the small pileup peak is sufficient. In reality it may not be.
 - b. New version (-10) of PCM file.
8. Spectrum_display:

- a. The launch routine for fit_setup defaulted to tweek off. Now it defaults to tweek=1 to show the “Adjust” tab in Fit Setup.
- 9. Spectrum_load_spectrum_increment:
 - a. (as used in add-by-detector plugin, or when combining results from cluster spectra extract/import)
 - b. Fixed so it also works for Maia detector spectra from Maia-Control too.

12 Nov, 2014

- 1. Fixed a bug in Association plot display:
 - a. Make_corr_tvb:
 - i. Error in estimating charge_per_pixel, typo xsize → ysize.
- 2. Image_eventcb: OnSelect_Image_Interp:
 - a. Make interp checkbox work for all elements together.
- 3. Multi-phase DA:
 - a. DA_evt:
 - i. Accumulate an average yield (actually 1/Y) with weighting just like used for events. Do this in new ‘da_accumulate8’.
 - ii. Save the new yield ave. in a new image in DAI. New version -48, saved as ½ size like error images.
 - b. Define:
 - i. Add new yield map.
 - c. Da_accumulate8:
 - i. Accumulate 1/Y using same weights as for events, and a normalization count ‘fny’. Invert and normalize back in Da_evt.
 - d. Read, write images:
 - i. Added code for yield maps, error map sized, version -48.
 - ii. Also fixed bugs that scaled a few maps with nsq*nsq, if compressed on input, when they should not (e.g. count-rate, pileup, dead-fraction).
 - e. Image_routines: analyze_image:
 - i. Use the new yield maps (take care with compressed size) to form ‘ayield’ average yield across regions.

16 Oct, 2014

- 1. Multi-phase DA:
 - b. DA_evt:
 - i. Use an image of phase proportions to weight use of a set of DA matrices
 - ii. Uses the new file ‘.mpdam’ for multi-phase DAM file, which points to the phase maps file and all DA matrix files (in ‘.correct’ file).
 - 1. Uses new read_da, that recognizes .mpdam file.
 - 2. Sets flag ‘mpda’ = 1
 - 3. Note /mpda mode is mutually exclusive of XANES mode.**
 - iii. Reads phase image file (given in .mpdam file)
 - 1. Congrids it to fit compressed, offset image area
 - 2. Builds DA phase weights array in ‘build_DA_phase_weights’
 - iv. Sort with weighting between DA matrices set by the phase maps at XY pixel, using new version of ‘da_accumulate7’.
 - v. Save ‘.mpdam’ file-name in DAI instead of DA matrix file-name.
 - vi. Choose one of the DA files for MDL (still use original?)
 - c. Read_DA:
 - i. Calls ‘read_mpdam’ to read correct file.
 - 1. Get ‘read_correct’ to do some retries nearby of matrices if not found.
 - ii. Dangerous as it depends on paths stored in ‘.correct’ file.
 - 1. Need to add new ‘file_requester(/skip_if_exists, /translate, updir=3)’ to look locally and use translation tables.**

2. Call with /silent from da_evt to suppress file_requester(). If error return, then exit da_evt.
- d. DA_accumulate7 (Fortran):
 - i. Pass weighting table and all phase DA matrices
 - ii. Weight each DA matrix by fraction of weight for phase (at E, XY)
 - iii. Assumes table is already normalized over phases at each E, XY.
- e. Build_DA_phase_weights:
 - i. Build the [E,phase,X,Y] weighting matrix for use with phase DA event processing.
 - ii. It will be compressed to manage size, and return an index array 'q' for access into the new table. Hence, to use the table:
 1. table[q[Eold], j, X/compress, Y/compress]
 2. **Keep size around 400-700 XY pixels only and 200 E**
 - iii. Call 'da_build_mpda_table (Fortran)' to build final table.
- f. DA_build_mpda_table (Fortran):
 - i. Weight of anticipated events for each phase at each E in a pixel:
 1. table(k,j,ix,iy) = spec(k,j) * phase(ix,iy,j)
 - ii. Table is normalized to sum to 1 over all phases at each E, XY.
- g. Build_DA_fit: spectrum overlay
 - i. Use 'ayield' calculated for region in 'analyze_image', if present:
 1. Uses 1/Y averaging within each pixel
 2. arithmetic Y averaging across region.
 - ii. Form phase-weighted pure spectra, weighted by yields.

6 Oct, 2014

1. Fit All fail problem solved:
 - h. Found error due to NaN values in cIntensity corrections from array_yield', probably due to changing element/line list.
 - i. Array_yield:
 - i. Found that if element/lines list changes, while 'use_last' option is in use (i.e. and the common block 'wIntensity' values were in use), then can end up with NaN values for the 'rEff_all' array and hence 'wIntensity' due to some 'e_line' values disappearing (zero).
 - ii. Made sure that det_eff values are clipped to 'small' to cure values in 'rEff_all' and 'wIntensity'.
 - j. Pixe_initial:
 - i. Only mask off small element, if /use_last is set then make sure that the line list does not change.
 - k. Pixe_fit:
 - i. Pass 'use_last' for looping to pixe_initial.
2. Plugin still with old device calls:
 - l. Gain_trim:
 - i. Fix occurrence of old 'device_adc_offset' and change to new 'adc_offset_device'.
 - ii. Also argument is now '(*pspec).DevObj'.
 - m. Also: Spectrum: Apply Linear, Gain Trim (various), Linearize (various), and Test routines.
 - n. Also: other devices, misc file imports.

2 Oct, 2014

1. Maia, Fastcom devices:
 - o. Add IC_mode=2 in "get_import_list" method.
2. Define:
 - p. Add a new parameter "IC_mode" to /import to allow setting the default charge/flux mode when the flux select panel is shown.
 - q. Make default value 2, for w/ PV.
3. Spectrum_load_new:
 - r. Collect sensitivity from get_header_info.

- s. Run flux_select with: charge_mode=opt.ic_mode, and goto finish if (*p).error.
- t. Use header to select file-names for pileup and throttle and /skip_if_exists.
- 4. Flux_select:
 - u. Gather 'sensitivity' data too, as passed parameter.
 - v. Added parameter for charge_mode, passed from 'spectrum_load_new'.
 - w. The pars storage *p now has an 'error' variable. Test on return in 'spectrum_load_new'.
 - x. If preamp.pv blank, set to "Maia:scaler.FC0".
 - y. Decode passed sensitivity to set the droplists for val and unit.
 - z. Collect IC_val and IC_unit from Maia device 'flux_scan' as passed pars. Use these to set preamp.val and unit.
 - aa. Remove line that set preamp.pv to first in list.
- 5. Get_maia_details:
 - bb. Decode pileup.info and throttle.info in var_val1
 - cc. If scan.info not struct, then concat all as comment string.
 - dd. Return in struct extra: comment, pileup, throttle (, IC_sensitivity?)
 - ee. Return monitor Val as string array. Decode ENUM case.
 - ff. 'IC_sensitivity' uses a direct search for "A1sens_ ..." PVs, like in Maia device flux_scan method.
- 6. Read_maia_32_header:
 - gg. Seems to need n_buffer very large (15,000,000 x3) to see the sensitivity PVs.
- 7. Maia device:
 - hh. Get_header_info:
 - i. Return in self.header: sample, grain, title, pileup, throttle, IC_sensitivity
 - ii. Flux_scan:
 - i. test (nq eq 0) instead on (q[0] ne -1).
 - ii. Also decode head.IC.sensitivity when suppress=1. Pass back as IC_val and IC_vunit.
- 8. Base device:
 - jj. Add to header: sensitivity, pileup, throttle, sample, grain.
- 9. Window resize problem:
 - kk. Image, Spectrum:
 - i. Check if event.x,y change, else just return.
 - ii. Add new variables to pstate: size_event_x, size_event_y
- 10. Replace:
 - ll. Now able to replace to null ''.
- 11. Geopixe_library:
 - mm. Temporary use of Windows DLL file.
- 12. Get_throttle:
 - nn. Line 19: rebuild 's' from 'str' not 's' strings.
- 13. File_requester:
 - oo. Add a translation table (as tab group with bookmarks)
 - i. Checks for file match using file, then path+file, then trans table.
 - ii. BUT: for this to work, need to pass as 'file' the full original file-name.
 - pp. Add a updir=n option to also check up 'n' dir levels.
 - qq. Need to remove the /noconfirm option in future (did not get used anyway).
 - i. Removed ALL references to it in numerous routines in v7.1.
- 14. Modifications to routines to make use of File_Requester() translation table:
 - rr. Spectrum_load_new:
 - i. Pass full file for pileup and throttle, and use /skip_if_exists.
 - ii. Get file from extended header mp.pileup.file, etc.
 - ss. Spectrum_display_eventcb: Notify ('image-region-select'):
 - i. Pass full file-name for matrix, and path (as fallback)
 - ii. Use updir=3, /skip_if_exists, /translate
 - tt. Image_table_EVT: 'EVT' button
 - i. Pass full original (*p).source to file_requester and path (as fallback)
 - ii. Same for (*p).source2.

- iii. Linear, Pileup, Throttle: all pass full file-name to File_Requester()
15. New routines:
- uu. Fix_paths(): make sure a trailing separator is present.

22 Sep, 2014

1. Image routines: Analyze_image:
 - a. If (*p).type is 1 (fraction) or 3 (STIM energy) then use average rather than divide by fcharge, in both image region and traverse code. For (*p).type=1 may use error image array too.
 - b. Form average of phase proportions and add to results as "phase".
 - c. Form average of yield values and add to results as "ayield".
 - i. Average 1/Y, weighted by phases, within a pixel
 - ii. Average Y across pixels in the region.
2. Map_spec:
 - a. Allowed it to extend length of mapped spectrum to not lose channel data.
 - b. Need to worry about returned length now.
 - c. Corrected "spectrum_add" to account for this change in return.
3. Geopixe_parallel:
 - a. Fixed 'default_device' call and recompile.
4. Device import spectrum routines:
 - a. Fixed a few old/bug references to old device routines and recompiled all.
 - b. Copied all device src to release dir.
5. Da_evt:
 - a. If matrix file has extension ".mpdam" then do multiphase DA (set mpda=1)
 - b. Read .mpdam file (file-names for .phase.dai and .correct files)
 - c. Sub-region or compress phase map if needed.
 - d. Read all component DA matrix files, and re-map matrix spectra according to individual E calcs onto first cal.
 - e. Only read new matrix for changed 'beam_energy' if mpda=0
 - i. XANES and multi-phase cannot be done together (at present).
 - f. If mpda, then call da_accumulate6, and pass mp_matrix and phase image.
6. Da_accumulate6:
 - a. Forms weighted average of all 'mp_matrix' values, weighted by 'phase' proportions in that pixel. Will make it ~3x slower.
7. Build_DA_fit:
 - a. Uses the regions "ayield" values to calculate weighted sum of pre spectra for overlays.
 - b. If no "ayield" is provided, then form average Y assuming phase weighting only.
 - c. 1/Y

29 August, 2014

1. Source setup:
 - a. New window to set-up sources.
 - b. Added "Transmission" target anodes to "Tube_spectrum2" and "Source_tube_spectrum".
 - i. Use a different integral for "y", limited to thickness of anode, in calculation of "f" factor in tube_spectrum2.
2. Plot_yields:
 - a. Changed plotting of 'xsect' to be versus continuum energy.
 - b. Changed Y axis radio buttons to select appropriate X axis options.
3. Spectrum display:
 - a. Duplicate:
 - i. Duplicates all displayed spectra, and appends them.
 - b. Median, smooth, compress, add:
 - i. All operate on all displayed spectra
 - ii. Unless in 'highlight' mode, then just the highlighted one.
 - c. Image display:
 - i. Use "copy_pointer_data" in 'image_copy'.

- d. Copy_pointer_data:
 - i. Add copy of object using 'clone_object'
 - ii. If detect a name ending in "_DEVICE", then use 'clone_device_object', which also copies device data.
- 4. Fit_setup:
 - a. Sliders:
 - i. Make increments to tails, shift, spread larger to be more useful.
 - ii. Increase range of 'Shift' to ± 0.05 .
- 5. Low-stats filter plugin:
 - a. Make a version (called 'low-state IDL') that implements Fortran as IDL code for debugging/testing.
- 6. Backgrounds:
 - a. Bauer:
 - i. Adjust width to be consistent with SNIP.
 - b. Bauer & SNIP:
 - i. New one takes maximum of both Bauer and SNIP, which seems good for XRF.
 - c. All (Bauer, SNIP, combined):
 - i. No median to start with, but end with a small median filter.

22 August, 2014

- 1. Continuum sources:
 - a. Define:
 - i. Added /source option to build a source definition/parameters struct.
 - b. Tube_spectrum2:
 - i. Generates an X-ray tube spectrum per dE bin (ph/mA/keV/sr/s)
 - ii. Can use Z, formula or Galinstan index
 - iii. Returns a list of all characteristic lines
 - c. Source_tube_spectrum:
 - i. Generates an X-ray tube spectrum per dE bin (ph/keV/sr/s)
 - ii. Uses formula and /weight option
 - iii. Returns a list of all characteristic lines
 - d. Make_tube_spectrum:
 - i. Can build a test source struct. Save this using 'write_source'.
 - e. Write_source, Read_source:
 - i. Write/Read a source struct to disk file.
 - f. Excillum_tube_spectrum:
 - i. Old form of 'source_tube_spectrum' that accepts Z and Galinstan=n
- 2. Continuum yields (see "Continuum spectra.docx"):
 - a. Calc_slices2:
 - i. Re-write 'calc_slices' to go on doubling slices for photons.
 - ii. Change maximum slice to 10 mg/cm² for ions, larger (500 mg/cm²) for photons. See lines "**** 21/8/14 CGR"
 - b. slow_beam:
 - i. NOTE: Large maximum slice thickness of up to 200 mg/cm² (which in practice probably limits to around 7 mg/cm²) caused the slowing down integration to stop at around 1 MeV, which may not be low enough for some PIXE applications. Change to 'calc_slices2' will help.
 - c. 'Calc_cross':
 - i. Does whole passed E vector in continuum mode now.
 - d. 'Calc_yield':
 - i. In continuum mode, weights spectrum by 'xsect' versus energy, and scales by 'branch_ratio', which changes with half-slice now.
 - ii. Returns differently in continuum mode:
 - 1. branch_ratio[k,i,n] for line 'k', element 'i', half-slices 'n'
 - 2. xsect[i,e] across continuum spectrum 'e'

- e. Geo_yield2:
 - i. Add 'beam' argument for the new continuum beam struct
 - ii. Changed default slice thicknesses to (i) 0.05 mg/cm² for photons, (ii) 0.02*e_beam for ions.
 - iii. For photons this was 0.05*e_beam, which could be as large as 1 mg/cm², which is far too coarse.
 - iv. NOTE: This will cause some XRF yields to change, as this slice was probably too coarse.
 - v. Add 'harden_beam' call to harden 'beam' spectrum versus half-slice.
 - vi. Get_lines' now takes 'beam' arg
 - vii. After 'get_lines' now weight relative intensities calculated as a function of beam energy in 'init_xrf_lines' with changing spectrum to calculate 'branch_ratio' versus half-slice.
 - viii. Call 'calc_cross' differently for continuum and mono/ion. Pass it 'beam' and 'spectrum.E' in continuum mode, so it returns 'xsect' versus photon energy.
 - ix. Speed issues:
 1. Needed to re-write 'calc_slices' to make it more efficient in XRF mode.
 2. Needed to call 'init_xrf_lines' only once per energy. This would be helped by keeping continuum spectrum small (<=500).
- f. Geo_array_yield:
 - i. Move to passing 'beam' struct as primary beam specification.
 1. Need to pass this from GUI next.
 - ii. Pass 'beam' to 'geo_yield2' and pass 'spec' between them with /reenter set.
- g. Make_peaks:
 - i. Change routine for building yield calculation results struct.
 - ii. Changes needed wherever it is used:
 1. Correct_lines, Select_element_lines, Append_escapes, Correct_da_loop, Calc_da_merge_scatter, Pixe_initial, Pige_initial, Sum_peaks, Layer_setup, Results_properties,
- h. Read_yield, write_yield:
 - i. Read/writes whole 'beam' source struct – **new version -8**.
- i. Layer_setup:
 - i. Add "Photons (continuum)" to beam options, which then maps OUT Z,A, charge, and IN a source file widgets and "New" and "Load" buttons.
 - ii. Load – loads from an existing .source file.
 - iii. Read/writes LCM file, with source file-name – **new version -5**.
3. Fitting:
 - a. Pixe_fit:
 - i. Add 'beam' to results, NOT into 'ryield' struct.
 - b. Read/write fit results:
 - i. **New version -14**
 - ii. Read/write source struct, using write_source/read_source routines.
 - iii. Beam struct is NOT read/write if continuum=0.
 - c. Result_properties:
 - i. Changed call to new 'make_peaks'
 - ii. Changed use of 'geo_yield2' to 'geo_array_yield'
 - iii. Added 'beam' to 'geo_array_yield' call
4. Implementation:
 - a. See notes in "Continuum spectra.docx".
5. Testing:
 - a. Compared 6.6r with this new 7.1a version.
 - b. Fits (refits) to PIXE (Donuts2x), XRF (single; pottery), XRF (Maia; rock2).
 - c. Yield calculations: PIXE, XRF, XRF (continuum) only using yields as "mono"
 - d. Results: Apply: filter, old yield, scaling.
6. New version = 7.1a

21 July, 2014

7. OM DAQ read bug with version 11
 - a. Read_om_header:
 - i. Rundata version 11 'sample' should have 15 layers.
 - ii. Offset to spectrum size 'ncalsize' is now 3, which effects 'escale' code to correct for compressed spectra.
 - b. Read_setup:
 - i. Zero 'lmf_charge'.
 - c. Read_buffer:
 - i. Increment 'lmf_charge'.
 - d. Spec_evt:
 - i. Test for 'OM_DAQ_DEVICE' does not use 'suppress' = 0
 - e. Spectrum_load_new:
 - i. Fixed error test on final post-import header read.

29 June, 2014

8. Gnumeric 'variable' error in 'chop_element'
 - a. Only shows as an error in geo_yield(2) because it's the first error catch.
 - b. Added "COMPILE_OPT STRICTARR" to chop_element and decode_formula, decode_radical.
9. @null_spectrum:
 - a. Replace with define(/spectrum), only if max_common parameters not used.
 - b. Append_da_fit, da_trav_evt, da_xstep_trav_evt, fit_overlay_pure, get_spec, get_text_spec, analyze_image, make_back, read_da_as_spec, spec_evt, spec_xstep_evt, spectrum_do_curve_fit, spectrum_background, time_amp_spec
10. @null_image:
 - a. Replace with define(/image), only if max_common parameters not used.
 - b. If so, make sure explicit common is used.
 - c. read_pnc_1D_scan, cut_evt, da_evt, cut_xstep_evt, da_xstep_evt, read_geopixe_image.
11. ftp_connect:
 - a. Added line to set PASV mode before USER.
 - b. Seems to fix problem with update from DESY on Linux.

27 June, 2014

1. File_requester
 - a. Add "Find" field to search for a file by pattern, then set "Path".

19 June, 2014

1. Get_throttle
 - a. Cope with both single channel per line, and "[]" form.
 - b. And also a range form "[n-m]".
2. Throttle3_select
 - a. Print without erroneous white space in Kandinski variable string.

15 June, 2014

1. Image_table_eventcb:
 - a. Notify: 'image-results-clear':
 - i. Remove test on (*pstate).p to enable 'clear_image_table_table' to be executed to clear table. This line was NOT in version 6.6 ??
2. Red colour blind palette:
 - a. Load_spec_colours:
 - i. New switch to enable/disable Red colour blind mode
 - ii. Save this state in a common for any calls to 'load_spec_colours' without the switch set.

- iii. Change red to a full blue colour.
 - b. Image:
 - i. Add menus to enable/disable Red colour blind mode.
 - ii. Change both image colour table to blues, and also spec colours.
 - iii. Notify both 'image-display' and 'spectrum-display'.
 - iv. Pass 'spectrum-display' notify to spectra and image clones get changes to attached spectra too.
 - v. Add 'spectrum-display' notify to image clones set-up.
- 3. File_requester:
 - a. Replace all instances of 'dialog_pickfile' except in some test routines
- 4. Spectrum Plugins in Highlight mode:
 - a. In highlight mode, single spectrum plugins (no "*" in title) will only use the highlighted spectrum, and not loop through all.

15 June, 2014

- 5. Maia_Launch
 - a. Fix name of log fil:
 - i. Added test for no path on "log_file" from defaults, which means no "logfile" parameter was in Maia.conf file.
 - ii. It then builds a name based on conf path and DAM identity.

2 June, 2014

- 6. How to revert to IDL 7.1.1?
 - a. Recompile any SAV file that is needed back to 7.1.1:
 - i. plugin sav files.
 - b. How to check sav files?
 - i. Run "geopixe_sav_list" and search for later IDL sav file in output.
 - ii. Check previous output of geopixe_sav_list.

23 April, 2014

- 7. get_aps_header:
 - a. The call to 'read_aps_lst_header' had the old device arg, which has now been removed.

7 April, 2014

- 1. ftp_post:
 - a. The FTP commands fail to work on Massive. Adding a <cr> to end of 'cmd' in ftp_post seems to fix it. Make it the default for posts from Unix systems.

5 April, 2014

- 1. file_write_test:
 - a. IDL command 'file_test(d, /write, /dir)' fails to work on Massive SAN.
 - b. Modify to actually try writing a dummy file instead.

28 March, 2014

- 1. wizard_depth:
 - a. Changed Notify scheme to use a linked list, with a 'callback' routine name to execute in wizard between actions and a 'pnext' to point to next in list.

24 March, 2014

- 1. wizard_depth: (add these lines to 'wizard_depth.pro' as comments)
 - a. Modify: image, image_table, imageRGB, spectrum_display to accept notify 'wizard-action' to perform tasks for wizards, and return 'wizard-return' notify when done.
 - b. Page 1: depth Curve
 - i. Calculate Depth Curve:

1. Wizard:
 - a. Calculate depth curve using wizard_depth_calc and save the results to file and file widget.
- c. Page 2: Cal Spec
 - i. Form Outer, Inner Cals:
 1. Wizard:
 - a. Read E cal spectrum and Outer, Inner select files to form Outer and Inner E cal/selection files.
 - b. Save these to disk files and the file widgets.
- d. Page 3: RGB
 - i. Sort Outer, Inner Images:
 1. Wizard:
 - a. Notify “Sort EVT” of ‘inner-outer’ to sort Outer and Inner detector images passing Outer, Inner select in turn.
 2. Sort EVT:
 - a. Sort Outer and Inner detector images files in turn, and save them to DAI files.
 - b. Reply Notify, passing Outer or Inner DAI file-names in turn.
 - i. Reply Notify done.
 3. Wizard:
 - a. Set Outer, Inner DAI file widgets.
 - ii. Display RGB Depth Images
 1. Wizard:
 - a. Notify “Image” window of ‘load-image’ to load the “Inner” image file.
 2. Image:
 - a. Load the Inner” image file.
 - i. Reply Notify.
 3. Wizard:
 - a. Notify “Image RGB” of ‘compare-image’ to load the outer in compare mode, and pass element.
 4. Image RGB:
 - a. Load the “Outer” image in compare mode.
 - b. Set element droplists to first occurrence (Inner) and second (Outer) for Red, Green respectively.
 - i. Reply Notify done.
- e. Page 4: Spectra
 - i. Extract Sum Spectra
 1. Wizard:
 - a. ‘Notify “Image Regions” of ‘extract-individual’ to use current region# (defaults to #0 now).
 2. Image Regions:
 - a. Set “individual” mode, sort spec_evt for all detectors.
 - i. Reply Notify, return file-name
 3. Wizard:
 - a. Notify “Spectrum Display” of ‘sum-spectra’, passing “Outer” .select.csv file and output file-name.
 - b. Notify “Spectrum Display” of ‘load-spectra’, passing original spectrum with all detectors.
 - c. Notify “Spectrum Display” of ‘sum-spectra’, passing “Inner” .select.csv file and output file-name.
 - d. Notify “Spectrum Display” of ‘load-spectra’, passing names of Outer and Inner sum spectra from above.
 4. Spectrum_Display:
 - a. Select only “Inner” detectors, delete others, add to form sum spectrum, save as “-inner.spec” file.

- i. Reply Notify, return label as qual2
 - b. Re-load all detectors from region sort spec file.
 - i. Reply Notify
 - c. Select only “Outer” detectors, delete others, add to form sum spectrum, save as “-outer.spec” file.
 - i. Reply Notify, return label as qual2
 - d. Re-load “-outer.spec” and “-inner.spec” files.
 - i. Reply Notify.
- 5. Wizard:
 - a. Responds to each Notify return from Spectrum Display to set outer, inner file-names in widgets.
- ii. Fit Results:
 - 1. Wizard:
 - a. Notify “Spectrum Display” of to re-load Outer, Inner in append mode.
 - 2. Spectrum_Display:
 - a. Re-load “-outer.spec” file.
 - b. Re-load “-inner.spec” file and append to Outer to display both.
 - i. Reply Notify.
 - 3. Wizard:
 - a. Notify “Spectrum Fit” of ‘fit-all’ to fit both Outer and Inner spectra, as loaded in Spectrum Display.
 - 4. Spectrum Fit:
 - a. “Fit All” to both outer, inner spectra as loaded.
 - i. Reply Notify.
 - 5. Wizard:
 - a. Notify “Spectrum Display” of ‘outer-inner-ratio’ to lookup Outer and Inner areas.
 - 6. Spectrum_Display:
 - a. Lookup Outer and Inner areas in fit results.
 - i. Reply Notify, passing area, error for outer, inner.
 - 7. Wizard:
 - a. Form ratio, error and determine depth from Depth Curve.
 - b. Add these results to the results table.
- f. Page 5: Results
 - i. Export results:
 - 1.

21 March, 2014

- 1. evt_start:
 - a. Fixed erroneous use of (*pstate).sort_mode with (*p).sort_mode.
- 2. maia_device:
 - a. Call to new ‘maia_accumulate_dtfx’ was passing xrange2,yrange2 as array dimensions, which is wrong for a Windowed sort. Changed to passing n_elements() applied to dims of ‘Flux’.
- 3. tv_q_mask:
 - a. For better Exclude mode shading, use ‘rebin(/sample)’ when zoom > 0.
- 4. ‘S pixel’ shape:
 - a. ‘Image’, ‘onbutton_image’: added code to handle single pixel selection.
 - b. Analyze_image: need to treat single *pq differently (offset by -0.5).
 - c. ‘read_regions’, ‘write_regions’: add type 11 (S pixel).
- 5. spec_evt:
 - a. ‘centre[]’ extended for centre handle on all 12 shapes.
- 6. spectrum_add:
 - a. Make sure any not ‘show’n spectra are also freed after add.
 - b. Fix error on obj_valid(DevObj) test (could NOT work with this).

28 February, 2014

1. Maia_setup:
 - a. Scepter apply:
 - i. Clip Trim and Thresh to be \geq zero after arrow tweaks.

19 February, 2014

1. Bauer background algorithm plugin
 - a. Based on the Bauer background used in AES (H.E. Bauer, "A fast and simple method for background removal in Auger electron spectroscopy", Fresenius J Anal Chem 353 (1995) 450-455).
 - b. Use 3*passes as width term "Delta".
 - c. Use 'low_stats' filter, as in SNIP, and allow Boost.

13 February, 2014

1. Added Delta pXRF spectrum import to Generic Device object:
 - a. get_delta_ascii_spec:
 - i. Read Delta pXRF CSV file, including headers lines
 - b. Generic device:
 - i. Add the Delta ascii read to import list
 - c. Spectrum_detail:
 - i. Change code to use 'obj->title()' to retrieve device title, like in image_detail.
 - d. List_device_objects():
 - i. Add /include_generic option to add to names and title.

2 February, 2014

1. Corrected error associated with 'scale' in results Properties':
 - a. apply_properties:
 - i. Add 'deadtime_correction' (like dt_corr in pixe_fit) to line:

```
(*p).conc = (*p).scale * (*p).deadtime_correction * float( (*p).area / yield)
```

20-23 December

Started work on DAQ raster generation, and to finish off the scan list and editor. Still will need to deal with ellipse shapes and the fly-scan versus step-scan better (see "scan control" doc).

1. Build_raster_list2:
 - c. Builds raster.step[] list for zig-zag style raster **step-scan**.
 - d. Determines fast axis to prefer DAC axis, and move to centre with stage if a DAC is used to try and cancel calibration errors for the DAC scaling.
 - e. Assumes that DAC axes are calibrated to "mm" scale too
 - i. will need to set "deflect.axis[].scale" and "units".
 - f. Method:
 - i. Move to origin as 'abs' first.
 - ii. If DAC, move stage on the same direction 'rel' to centre, then move the DAC back 'abs' same amount.
 - iii. Move along fast scan direction line in steps, using time, charge, photons min/max settings.
 - iv. If 100x overscan enabled, then loop
2. Build_raster_list3:
 - g. This one assumes a constant velocity **fly-scan**, and a single step spans the entire fast direction scan line. Ignores 'charge', 'photons' min/max options.
 - h. Estimates velocity from line time estimate ($vel = sizeX / (T_{pixel} * nx)$).
3. Scan_list:
 - i. Start:
 - i. Look for "START" or first "ON" active mode in *plist.

- ii. Build_raster_list3 for first.
 - iii. set local active=1 and raster.on=1
- j. Notify:
 - i. Responds to 'scan-sequence-next' to:
 - 1. increment index (skip OFF, etc.)
 - 2. build_raster_list3 again
 - 3. Notify, 'scan-sequence-started' again.
 - ii. if all done, Notify 'scan-sequence-complete'
- 4. Daq_launch:
 - k. Timer: ('setup-daq-button')
 - i. get "raster.status" from Klee to set (*pm).control.status.raster_status
 - ii. monitor status not "run" or "pause" when active is on
 - 1. if not error, then Notify 'scan-sequence-next'
 - l. Notify "":
 - i. Responds to 'scan-sequence-started' to:
 - 1. set local active=1 and raster.on=1
 - ii. Responds to 'scan-sequence-complete' to:
 - 1. set local active=0 and raster.on=0

15-18 December

Start conversion from old objects index, which was global to all and in common, to new local instances of objects in images, regions, spectra, select, import and the Sort EVT windows.

- 5. New device objects routines:
 - m. define_devices:
 - i. restore device object SAV files
 - ii. build list of device names and titles
 - iii. open_device_objects
 - iv. save name, title, beam_type in common
 - v. build import list
 - vi. free objects at end
 - 1. do not store obj array anymore
 - vii. remove 'default_device' keyword
 - n. open_device_objects
 - i. restore SAV files
 - ii. get 'name' and 'title' for each
 - iii. also do 'beam_type' for each
 - iv. what about generic device?
 - v. create new objects, return as vector, as it does now
 - o. find_device_objects
 - i. only look for SAV files, as in 'open_device_objects2'
 - p. list_device_objects, title=title, beam_type=beam_type
 - i. return names (and title, beam_type) of all device objects
 - ii. does this also append the generic device?
 - q. list_device_imports
 - i. return vector of 'opt' structs for all imports
 - ii. each 'opt' have device name in it
 - r. free_device_objects
 - i. accept an obj (or array)
 - ii. free all.
 - s. clone_device_object
 - i. make a copy of dev object, and copy options too.
 - t. adc_offset_device():
 - i. replaces device_adc_offset()
 - u. adc_list_device:

- i. replaces device_list
 - v. find_id:
 - i. search for a widget by uname
- 6. Obsolete routines:
 - a. Those that used old device index, not used anymore ...
 - i. Replaced by obj->method(), (where method name is quoted)
 - 1. device_adc_offset
 - a. related to start_adc() method
 - b. use new 'adc_offset_device()'
 - 2. device_big_endian
 - 3. device_extension
 - 4. device_object_name
 - 5. device_multi_char
 - 6. device_multi_files
 - ii. Replaced by new versions (see above):
 - 1. device_import_index
 - a. use new: list_device_imports
 - 2. device_list
 - a. use new: adc_list_device()
 - 3. device_object
 - a. use Obj instance, via obj_new(name)
 - iii. Not needed anymore, use 'obj' instance:
 - 1. device_object_index
 - 2. set_object_device
- 7. Routines with changed pars:
 - a. device_index_from_old_index:
 - i. returns new name, but not device index
 - b. get_header_info:
 - i. pass Obj, not index
 - c. flux_scan:
 - i. pass Obj, not index
 - d. flux_select:
 - i. pass device=Obj, not index
 - e. read_event_buffer:
 - i. pass 'obj' instead of 'device' index
 - f. read_aps_lst_header:
 - i. drop 'device' arg, unused
 - g. device_specific, read_buffer:
 - i. pass device object, not index
 - h. scan_dir_evt:
 - i. pass 'obj' rather than 'device' index
 - i. evt_start_image_increment2:
 - i. pass 'obj' rather than 'device' index
 - j. batch_sort:
 - i. pass 'obj' rather than 'device' index
 - k. make_ystep:
 - i. pass 'obj' rather than 'device' index
 - l. time_amp, time_amp_evt:
 - i. pass 'obj' rather than 'device' index
 - m. 'device' notify:
 - i. from EVT (to Batch sort)
 - ii. pointer to DevObj passed
 - n. spectrum_load, spectrum_load_new:
 - i. pass 'opt' not img
 - o. da_evt:

- i. pass device name as 'device'
- p. spec_evt:
 - i. pass device name as 'device'
 - ii. clones obj for each spectrum
- q. da_xstep_evt:
 - i. pass device name as 'device'
- r. da_trav_evt:
 - i. pass device name as 'device'
 - ii. clones obj for each spectrum
- s. spec_xstep_evt:
 - i. pass device name as 'device'
 - ii. clones obj for each spectrum
- t. cmit_evt, cmit_xstep_evt:
 - i. pass device name as 'device'
- u. xanes_da_evt:
 - i. pass device name as 'device'
- v. xanes_single_da_evt:
 - i. pass device name as 'device'
- w. import_select:
 - i. returns struct with 'opt', but no 'img', 'device'

1.

8. Device object changes needed

- a. New approach for Sort Options and Obj
 - i. Store Obj in top base uvalue of Sort Options widgets
 - 1. store 'self' as uvalue in the top base in Render
 - 2. set uname to 'obj-ref-here'
 - ii. event handler:
 - 1. event.handler will always point to the top base in Render
 - 2. check that uname is 'obj-ref-here', and obj valid
 - iii. Realize routines:
 - 1. search up from wWidget for top base using:
 - 2. ObjBase = find_id(wWidget, search='obj-ref-here')
 - 3. extract obj from uname
 - 4. check that it is a valid object (see examples in Maia device)
 - 5. render routine: store self and 'obj-ref-here'
 - 6. realize routine: use find_id to get 'obj'
 - 7. event routine: use event.handler
 - 8. Applied this approach to:
 - a. Maia, Fastcom MPA3, DAQ, NSLS MCA

9. EVT

- a. Device objects now in *p
 - i. *p has 'device' index, as before
 - ii. *p now has pDevObjList, a pointer to a device object array
 - iii. DevObj = (*(p).pDevObjList)[(p).device]
- b. Add new Obj reference pointer "DevObj".
 - i. Remove "device" index, "device_name" and "devpars" pointer
 - ii. Do not store devpars in this struct (keep local to device Obj)
 - 1. No need to copy (set_options) devpars into device before sort or write
 - 2. No need to read (get_options) into devpars from device Obj.
 - 3. Read triggers Device to read pars internally.
- c. use 'open_device_objects' to create obj vector for all
 - ii. save in pointer: (p).pDevObjList (pointer to Obj array)
 - iii. use methods from obj->name(), obj->title(), not common, in case new ones added or removed.
- d. Device droplist/index changes:

- iv. (*p).device = event.index
- v. local (e.g. top of event routine):
 - 1. DevObj = ((*p).pDevObjList)[(*p).device]
- e. From DAI:
 - i. creates new Image struct, including new dev Obj
 - 2. devpars set internally on image read
 - ii. search for the same Obj name in list, sets (*p).device index to this obj
 - 3. new Image dev Obj should replace Obj in list (free old)
 - a. find obj->name() in ((*p).pDevObjList) = i
 - b. (*p).device = i
 - c. free object ((*p).pDevObjList)[i]
 - d. ((*p).pDevObjList)[i] = Obj
 - iii. old device_object((*p).device) will be replaced by DevObj for methods:
 - 4. old: (device_object((*p).device))->multi_files()
 - 5. new: (*p).DevObj->multi_files()
- f. Exit:
 - i. NOTE: EVT pars are stored in parent Image.
 - 6. freed (badly) in 'free_image_state', missing pointers and objects
 - a. free added for: 'root', 'pic_list', 'layout', any objects
 - ii. The 'pDevObjList' objarr and 'device' index are stored in *p (not *pstate):
 - 7. *p: YES
 - a. use this for 'device' index, as now
 - b. free *pevt when Image closes
 - c. index remembered between openings of EVT
 - i. but index shared between multiple EVT?
 - iii. free Obj ref in Image struct in free_images
 - iv. TAKE CARE:
 - 8. Only free these DevObj's in the list when parent of EVT is closed, via freeing *pevt.
- 10. evt_start:
 - a. Pass 'device_name' and 'devpars': to da_evt, etc.
 - i. Retrieve 'devpars', "device_name" just for the call.
 - 9. Can be added to pars for cluster, using 'stringify'.
 - ii. Create local Obj for this device_name in DA_evt, spec_evt, etc.
 - 10. Set 'devpars' into this local Obj, using set_options()
 - 11. Remember to free local Obj if it is not passed back (e.g. cluster mode).
 - iii. Makes sense for cluster execute in another process
 - b. Makes sense locally too, as we need to create an Obj ref that will be linked to the output image struct and passed back (non cluster mode).
 - c. use "DevObj" in place of "device_object((*p).device)", as above
 - d. replace "device_extension((*p).device)" with "DevObj->extension()"
- 11. DA_evt, spec_evt (and xstep, trav versions):
 - a. Pass 'device_name' and 'devpars' struct:
 - i. Create local 'DevObj' and use 'devpars' to set internal options
 - b. At end, if return pointer to image, leave DevObj referenced there.
 - i. If not (e.g. cluster), then Free DevObj (as part of free_image).
 - c. spectra: each spectrum in array uses a clone of obj
- 12. Batch sort:
 - a. Get notify of new device Obj, but do NOT free it when batch closes. It is 'owned' by Sort EVT window.
- 13. import_select:
 - a. Uses new 'list_device_imports' and 'list_device_objects' without the need to create objects.
 - b. Returns: 'opt' and 'device' name selected.
- 14. Spectrum_load_new:
 - a. Create obj based on device name.

- b. Need to call flux_select using the DevObj ref, not index.
- 15. flux_select:
 - a. pass in DevObj, not the index.
 - b. can we just go straight to the new flux_scan method here?
- 16. flux_scan:
 - a. pass in DevObj, not the index.
 - b. can we use flux_scan method instead?
 - i. only difference is the 'dir_mode' for XANES?
 - ii. Is this still being used? Yes, for EXAFS.
 - iii. Similarly for 'get_header_info', so keep both.
- 17. write_spec, read_spec:
 - a. read/write the same (device name, devpars via methods)
 - b. handle (*p).DevObj
 - c. write device name once for all spectra (assumes the same device)
 - d. reads once, but clones obj for each spectrum
- 18. write_geopixe_image, read_geopixe_image:
 - a. read/write the same (device name, devpars via methods)
 - b. handle (*p).DevObj
- 19. Spectrum:
 - a. fixed spectrum_display, events, routines for new DevPars
- 20. Image:
 - a. fixed image, events, routines for new DevPars
 - b. add obj = (*p).DevObj
 - c. chase through new obj-> stuff
- 21. image_routines:
 - a. analyze_image:
 - i. make a clone of image DevObj for each region created
 - ii. same for each spectrum in traverse (since spectra can be deleted)
 - iii. these regions (spectra) can be freed independently
- 22. read_region, write_region:
 - a. Make a clone of image DevObj for each region created
 - b. The read is tricky as we need to cater for the following version ranges:
 - i. -9 to -18:
 - 1. no device info, assume MPSYS_DEVICE
 - ii. -19 to -20:
 - 1. Create obj (assume Maia)
 - 2. Read explicit devpars struct (Maia terms)
 - 3. Set 'devpars' using 'set_options' method
 - 4. Also read old device index, create obj, (kill old 'obj')
 - 5. Set previous 'devpars' into this 'obj'
 - iii. <= -21
 - 1. Read device-name first, create obj
 - 2. Get options into obj using 'read_options' method
- 23. image_table, image_table_event:
 - a. add obj = (*p).DevObj
 - b. chase through new obj-> stuff
- 24. spectrum_select, spectrum_select_eventcb:
 - a. add obj = (*p).DevObj
 - b. chase through new obj-> stuff
- 25. maia_launch, maia_setup:
 - a. add DevObj to *pm struct
 - b. initialize images and spectra using image.DevObj = clone_device_object()
- 26. daq_launch, daq_setup:
 - a. add DevObj to *pm struct
 - b. initialize images and spectra using image.DevObj = clone_device_object()

11 December

Broadly have these categories of charge/flux collection for ion beam and synchrotron XRF applications:

27. charge direct

- a. integrate charge directly (IC mode = 0)
 - i. no flux map
 1. just enter a total charge for map, assume evenly distributed
 - ii. flux map (is charge map)
 1. charge is total of flux map (conv=1.)
 2. make sure 'read_buffer' will convert to uC for flux map/sum.
 - iii. IC mode = 0, no conv is visible or needed.
 1. Store conv = 0.0
- b. charge indirect via using ion chamber or other detector (IC mode = 1,2)
 - i. collect flux map
 - ii. Explicit conv factor, charge = total(flux) * conv

Modifications to routines:

1. read image:
 - a. if IC mode = 0, then
 - i. use read Charge (even if zero)
 - ii. leave conv=0.0
 - b. if IC mode > 0, then
 - i. read conv, PV, etc.
 - ii. set Charge to conv*flux (if charge = 0)
 - iii. if flux map found, set conv = charge/flux
2. Sort EVT template (from DAI):
 - a. if IC mode = 0, then
 - i. use read Charge (even if zero)
 - ii. leave conv=0.0
 - b. if IC mode > 0, then
 - iv. read conv, PV, etc.
 - v. set Charge to conv*flux (if charge = 0)
 - vi. if flux map found, set conv = charge/flux
3. DA_evt:
 - a. No flux_IC defaults to mode=0, conv=0
 - b. Fixes charge, flux, flatten in 'image_correct_flux'
4. image_correct_flux:
 - c. IC mode 0
 - vii. force conv = 0.0
 - viii. if flux map present, assume it's charge
 12. charge = total(flux)
 - d. IC mode 1,2
 - ix. if flux map present, charge = conv * total(flux)

9 December

2. spec_evt:
 - a. dead_fraction mean() only over active detector channels.
 - b. normalize dead_fraction to obj->get_total_time()
3. devices:
 - a. read_setup:
 - i. initialize self.dwell_total to 0.0 (for spectra) on /first
 - b. read_buffer:
 - i. accumulate dead_fraction for spectra mode
 - ii. accumulate count in self.dwell_total
 - iii. also fix dead_fracton and flux in image mode.
 - c. Make changes for:

- i. NSLS NetCDF, NSLS MCA, NSLS HDF
 - ii. GSE-CARS, Hasylab, ESRF
 - iii. NSLS NetCDF: returns lines of data times n_detectors, so must accumulate stripes in dead_fraction and flux. Used rebin before, which might failed with an odd number of Y rows and ycompress. Now uses smart_config (which can handle odd Y rows, but falls back to rebin when it can for speed).
- 4. base_device:
 - a. Now does not do base_device__define so many times (called from define_devices).
- 5. image_details:
 - a. Fixed ADC # offset for devices that don't start at 0.

7 December

1. Added device specific X,Y ADC controls to the Fastcom MPA3 device:
 - a. Fastcom MPA3 device:
 - i. Uses version tests in read_options.
 - ii. Add the following routines to service X,Y ADC selection:
 1. read_options, write_options – file I/O
 2. options_legend – to add to the properties windows
 3. onrealize... - to realize XY ADC widgets
 4. event routine
 5. render_options – to add widgets for XY ADCs
 6. set_options, get_options
 - b. Base device:
 - i. From version -46 of images, base device will read/write a device version of 0L.
 - ii. Remove set_versioning from Init method
 - iii. write_option:
 1. write version 0L
 - iv. read_option:
 1. read version, nothing else if version is 0L
 - v. write version 0L
 - c. Maia device
 - i. Remove set_versioning from Init method.
 - ii. Also fix DAQ device draft.
 - d. Read_geopixe_image:
 - i. Add test for Maia device. Use version = -46 as start for versioning of non-Maia devices.
 - e. read/write regions:
 - i. Add test like image, for version = -25 as start for versioning of non-Maia devices.
 - f. Define:
 - i. Add version to image struct.
 - g. Image_details:
 - i. Add showing image version number.

5 December

A significant upgrade of the device drivers, mostly to fix Maia (as a better template for new devices), but also to fix over use of special parameters only really used in MPsys device (e.g. xstep args).

1. device_specific (which calls device 'read_setup' method):
 - a. Don't show special keywords explicitly (e.g. ystep), but pass them using _ref_extra mechanism.
 - b. NOTE: this means that devices that use these need to test for undefined and set a default value (e.g. some not passed from non-xstep routines).
2. read_buffer (which calls device 'read_buffer' method):
 - a. Don't show special keywords explicitly (e.g. ystep, xstep..., by_odd, toggle...), but pass them using _ref_extra mechanism.

- b. This includes new 'veto' vector.
 - c. NOTE: this means that devices that use these need to test for undefined and set a default value (e.g. some not passed from non-xstep routines).
 - d. 'ste', 't', 'n' set to zero ('multiple to -1L) scalar before calling device read_buffer method.
 - e. If these are still scalar afterwards (not used in device), then set them to zero (or -1L for multiple) UINT (LONG for multiple) vectors of length 'n'.
3. Maia device:
 - a. read_setup:
 - i. Remove ystep keyword.
 - ii. Don't use (*self.dwell)[] for dwell map – more efficient to use maia_dwell in common.
 - b. get_dwell:
 - i. Now uses common maia_dwell.
 - c. read_buffer:
 - i. Remove ystep and all xstep keywords and comments. Keep the ones unique to Maia, such as by_odd, by_even support_even_odd.
 - ii. New variables (in common) to track last PV flux and time returned from read_buffer: maia_last_flux, maia_last_time.
 - iii. Now create vectors for x1,y1,e,t,ste,veto of length n_events
 1. created once only in common (maia_e, maia_t, maia_x, ...).
 2. new vector 'veto' as an event veto (set =1 for pseudo events).
 3. with new one maia_fx[4,*] for storing flux, dwell as pseudo events from Fortran routine 'maia_384_events4'.
 4. with a high water mark of 'n' and 'good' for those with veto=0
 5. use 'veto'=1 to suppress other clipping, etc., so we don't need to shorten vectors in da_evt a few times using e=e[q].
 - iv. Shorten these vectors to make output ones (of length 'n') only once.
 1. any removal of events (odd/even, clear borders) just sets veto=1.
 - v. Remove all use of the 2D stripe image arrays maia_tt, maia_fc0, etc.
 - vi. Accumulate dead_time and flux, dwell in new Fortran routine: 'maia_accumulate_dtfx'.
 1. image mode:
 - a. accumulate both dead_fraction and flux in Fortran.
 2. spectrum mode:
 - a. accumulate just dead_fraction in Fortran, and flux in Maia device, using total of fx[0,*] in h/w mode or current Epics rate value in Epics mode.
 - vii. Update Fortran routines:
 1. maia_384_events4
 2. maia_accumulate_dtfx
4. maia_384_events4
 - a. New Fortran reader, with fx, veto vectors. Returns :
 - i. maia_last_flux, maia_last_time – last flux Epics PV values in buffer based on passed flux PV name.
 - ii. maia_last_energy – last energy Epics PV values in buffer based on passed energy PV name.
 - iii. maia_first_time – first time stamp of any kind in buffer.
 - iv. Remove all use of the stripe images arrays.
5. maia_accumulate_dtfx
 - a. New Fortran routine to accumulate flux and dead_fraction in Maia device
 - b. image mode: /image
 - i. Use pseudo events (veto=1) only.
 1. Accumulate flux from FC0, FC1 into planes 1,2
 2. Accumulate selected flux in flux array plane 0:

- a. flux_mode=0: set use Epics PV, assumed to be a rate time flux_scale, assumed to contain rough dwell.
 - b. flux_mode=1,2: accumulate using h/w FC0, FC1 counters.
 3. Accumulate Dwell.
 - ii. Use valid normal events (veto=0) only:
 1. Accumulate dead-fraction by pixel XY, scaled by DT calibration to μ s.
 - c. spectrum mode: /spectrum
 - i. Use valid normal events (veto=0) only:
 1. Accumulate dead-fraction by detector #, scaled by DT calibration to μ s.
6. MPsys device:
 - a. read_setup
 - i. Needs to test for undefined special option (ystep) and set default.
 - b. read_buffer
 - i. Needs to test for undefined special options (e.g. ystep, toggle_..., step_...) and set defaults.
 - ii. Make sure that both 'n' and 'good' return valid event count and length of x1,y1,e vectors.
 - iii. MPsys is the only device that keeps all the xstep, ystep keywords.
 - iv. MPsys ignores the new 'veto' vector, as well as 'multiple'.
7. Other devices:
 - a. read_setup:
 - i. Remove ystep keyword.
 - b. read_buffer
 - i. ***Make sure that 'n' and 'good' for most equal number of events returned in the x1,y1,e vectors.***
 - ii. Remove ystep and all xstep keywords and comments.
 - iii. Add comments about optional veto, t, multiple.
8. da_evt:
 - a. Use new 'veto' in call to read_buffer.
 - b. Test if 'veto' is still scalar afterwards (not used in device), then set it to zero UINT vector of length 'n'.
 - c. Any shortening of vectors (e.g. clipping, random subset) now just sets veto=1 for events to remove, without shortening the vectors.
 - i. q = index to good events
 - ii. count2 = number of good events.
 - d. Update Fortran routines:
 - i. da_accumulate5
9. da_accumulate5:
 - a. Accepts 'veto' and 'n' as well as 'good' (just use for test), in addition to the 'pu', 'nnpu', 'nn' that were already in da_accumulate4.
10. spec_evt
 - a. Use new 'veto' in call to read_buffer.
 - b. Test if 'veto' is still scalar afterwards (not used in device), then set it to zero UINT vector of length 'n'.
 - c. Any shortening of vectors (e.g. clipping, random subset) now just sets veto=1 for events to remove, without shortening the vectors.
 - i. q = index to good events
 - ii. good = number of good events.
 - d. Update Fortran routines:
 - i. spec_det_accumulate3
 - ii. spec_accumulate3
 - iii. hist_accumulate4
11. spec_det_accumulate3
 - a. adds veto test to previous spec_det_accumulate2
12. spec_accumulate3

- a. adds veto test
- 13. hist_accumulate4
 - a. adds veto test
- 14. da_xstep_evt
 - a. Use new 'veto' in call to read_buffer, as well as ystep keyword.
 - b. Test if 'veto' is still scalar afterwards (not used in device), then set it to zero UINT vector of length 'n'.
 - c. Any shortening of vectors (e.g. clipping, random subset) now just sets veto=1 for events to remove, without shortening the vectors.
 - i. q = index to good events
 - ii. count2 = number of good events.
 - d. Update Fortran routines:
 - i. da_accumulate5
- 15. spec_xstep_evt
 - a. Use new 'veto' in call to read_buffer, as well as ystep keyword.
 - b. Test if 'veto' is still scalar afterwards (not used in device), then set it to zero UINT vector of length 'n'.
 - c. Any shortening of vectors (e.g. clipping, random subset) now just sets veto=1 for events to remove, without shortening the vectors.
 - i. q = index to good events
 - ii. good = number of good events.
 - d. Update Fortran routines:
 - i. spec_accumulate3
 - ii. hist_accumulate4
 - iii. NOTE: spec_xstep_evt does not support 'by_detector' mode.
- 16. cut_evt
 - a. Use new 'veto' in call to read_buffer.
 - b. Test if 'veto' is still scalar afterwards (not used in device), then set it to zero UINT vector of length 'n'.
 - c. Any shortening of vectors (e.g. clipping, random subset) now just sets veto=1 for events to remove, without shortening the vectors.
 - i. q = index to good events
 - ii. count2 = number of good events.
 - d. Update Fortran routines:
 - i. cut_accumulate5
- 17. cut_accumulate5
 - a. adds veto test
 - b. simplify using test on 'multiple(0)'.
- 18. cut_xstep_evt
 - a. Use new 'veto' in call to read_buffer, as well as ystep keyword.
 - b. Test if 'veto' is still scalar afterwards (not used in device), then set it to zero UINT vector of length 'n'.
 - c. Any shortening of vectors (e.g. clipping, random subset) now just sets veto=1 for events to remove, without shortening the vectors.
 - i. q = index to good events
 - ii. count2 = number of good events.
 - d. Update Fortran routines:
 - i. cut_accumulate5

29 November

- 1. Da_evt:
 - a. Move Hist build to after filter on x1, y1 and col in range.
- 2. Make changes to GeoPIXE-source, according to:
 - a. **"Move GeoPIXE2 (develop) to GeoPIXE (stable).docx"**.

27 November

1. Copy GeoPIXE-source2 to GeoPIXE-source.
2. Make changes to GeoPIXE-source, according to:
 - a. “**Move GeoPIXE2 (develop) to GeoPIXE (stable).docx**”.
3. Version 7.0 in GeoPIXE-source2. Keep version 6.6j in GeoPIXE-source.

22 October 2013

1. Depth profiling (see doc file “Particle Depth Probing.docx”)
 - a. yield calculation
 - i. Fixed ‘layer_setup’, calculate event code.
 - ii. Fixed minor errors in ‘geo_array_yield’, ‘geo_yield2’.
 - b. depth profiles
 - i. ‘ratio_detector_yields’: New routine, drawing on example code in ‘pixe_fit’, to combine “inner” and “outer” detector selections to call ‘array_yield’ to calculate ratio of yields for elements from grid yield data.
 - ii. ‘depth_ratio_yields’: Pop-up to call ‘ratio_detector_yields’, called from Spectrum Display window menu.
2. Version 6.6j

17 October 2013

1. DAQ launch:
 - a. DAQ_setup:
 - i. Parameter set-up.
 - ii. DAQ_update routines updated for Klee parameters client(s).
 - b. DAQ_clients:
 - i. Klee parameters client (slow one not enabled yet).
 - ii. Blog clients updated.

8 October 2013

1. Missing ‘flux’ signals
 - a. Due to blank (*pm).IC.pv.name
 - i. Copied blank name to ‘blog_client_da2’ as ‘Maia_IC_name’, which caused maia_hw_counter to be not set.
 - b. New checks:
 - i. Maia-setup: test for non-blank pv name and set to first of plist if blank
 1. Test on Maia-setup open
 2. Test in rates timer in Maia-launch
 3. Test in apply imaging
 - ii. If get copied to Kandinski, it will get read back OK.
 - c. ‘blog_client_da2’:
 - i. Seemed to miss scaling by (*pf)[4] sensitivity to form ‘dflux’ when in ‘maia_hw_counter’ mode.
 - ii. Added a line to scale by (*pf)[4].
2. Version 6.6i

12 September 2013

1. Cuts images
 - a. Accumulate total counts, and ignore multiplicity scaling (1/xd)
 - b. Fix bug in evt_start_image_increment2 that forced global (post cluster stripes) flatten even if not set.
2. Align image plugin
 - a. Compare all images to first now, get absolute X,Y shifts
 - b. This fixes occasional error creep that became a drift of position.
3. shift_image_rows

- a. Correct edge behaviour to extend and not wrap/rotate.
- b. Note that full image 'shift_image' routine still uses wrap/rotate
- 4. Maia device:: read_buffer
 - a. Fixed xoffset of flux arrays in Window sort mode
 - i. used [xoffset:* ,0:dy] to select sub-set of 'Maia_' arrays in 6 calls to 'image_increment' routine.
 - b. Note that offset args for image_increment are ambiguous
- 5. Version 6.6h

19 August 2013

- 1. shapes rounding for zoom > 0 on small images zoomed out
 - a. Now do not round on display as shape is moved. So movement is nice and smooth, even at large zoom and pixelated images.
 - b. onbutton_image
 - i. use /fractional in call to pixel_to_xy to have float x,y positions in data
 - ii. do this both for Down and Move.
 - iii. maintain region shapes as floats
 - iv. Subtract half a pixel size (0.5*dx, 0.5*dy) to correct origin in Box origin display
 - v. *** This assumes that motor position at origin is in the middle of the pixel.
 - c. pixel_to_xy
 - i. add /fractional keyword
 - ii. if fractional, then clip to size-0.1 instead of size-1
 - d. zoom
 - i. add /fractional keyword
 - ii. if fractional, keep as float, then clip to 0 not 1 at low values
 - e. analyze_image
 - i. do not need to fiddle size-1 for polyfillv if shape includes at least size-0.5.
 - ii. Some old region shapes may be affected and miss rightmost or topmost pixels of image.
- 2. plot_images
 - a. Fix error with "absolute" so that switch to 'microns' below 0.3 mm range does not happen now, which upsets plot scales.
 - b. Still needs to be added to Load/ Save of plot options.
- 3. Flux flattening issues resolved:
 - a. Was caused by effectively applying raw flux flatten again to the attribute (FC) planes in post-cluster image reconstruction. Need to instead flatten to the attribute plane (selected for flux norm), which has been flattened in the cluster stripes to just remove final differences between stripes.
 - b. da_evt:
 - i. saves flux[* ,* ,0] as raw_flux
 - ii. builds dead_fraction and pileup-losses, saves to *pimg
 - iii. clips dead_fraction at 0.5 and pileup-losses at 0.9
 - iv. corrects flux[* ,* ,0] for both dead_fraction and pileup-losses
 - v. calls image_correct_flux w/ cluster_pass=0 (default)
 - c. evt_start_image_increment2:
 - i. call image_correct_flux with /cluster_pass
 - d. image_correct_flux:
 - i. Looks up index of any attribute used as flux, passes as 'use_FC_index'
 - ii. Calls image_flux_flatten (passes cluster_pass switch and FC index)
 - iii. Saves flux[* ,* ,0] to (*pimg).flux
 - iv. Saves raw_flux to (*pimg).raw_flux
 - e. image_flux_flatten:
 - i. flattens image to flux[* ,* ,0], and flattens flux[* ,* ,0] itself
 - ii. cluster_pass = 0
 - 1. flattens attribute (FC0, FC1) planes of image to raw_flux

- iii. cluster_pass = 1
 - 1. Uses the selected FC index attribute plane to flatten attribute images after cluster reconstruction of stripes.
- 4. Version 6.6g

15 August 2013

- 1. plot_images, plot_image_select
 - a. Add “absolute” option to show real lab coordinates in plots
 - b. still needs to be added to Load/Save options files.
- 2. Added to NSLS NetCDF device driver
 - c. New Mark Rivers routines (looks the same, but may have changed).
 - d. Pick up scan details from “_attributes.sav” files.
- 3. Version 6.6f

27 July 2013

- 1. pixe_fit, pixe_cleanup
 - a. Use /silent to suppress all but first PU warning
- 2. Added some diagnostics to the YLUT file access/paths.
- 3. Version 6.6e

6 July 2013

- 1. Spec_evt
 - a. By_even, By_odd – problem with using this with Ycompress not 1.
- 4. read_buffer
 - a. Add _ref_extra keyword passing between read_buffer and method it calls.
 - b. Use this to pass on new (Maia specific?) keywords
- 5. Maia device
 - a. Add _ref_extra keyword

6 June 2013

- 1. Maia_launch, Maia_setup, Maia_client_parameters, Maia_client_parameters_slow ...
 - b. Add Scepter Thresh and Trim adjustment arrow button.
 - i. Offset from (*pstate).scepter.thresh, and .trim
 - 1. (remember that Trim stored +ve, output -ve)
 - ii. uname = 'hymod-gaintrim-up', 'hymod-gaintrim-down'
 - 1. 'hymod-apply-gaintrim-adj', 'hymod-save-gaintrim'
 - iii. Stores tweaks in (*pstate).scepter.tweak.thresh.val, similarly for Trim.
 - iv. A flag “(*pstate).scepter.tweak.thresh.on” flags changes
 - v. Apply to Hymod uses the tweaks if flag(s) set
 - c. Add gaintrim T adjustment arrow buttons on Hymod panel
 - i. This adjusts local copy of T gaintrim offset: (*pl).channel[].trim.T.b
 - ii. Apply sends them to Hymod, Save, saves to disk as .gaintrim.time.var
 - d. **Order of gaintrim** in (*pm).channel was **wrong** (should be CSV index order)
 - i. This effects Hymod apply, read, write, update of these parameters
 - ii. Summary table showed gaintrim in wrong order
 - iii. Fixed all use of channel[].trim
 - iv. Affected: maia_client_parameters_slow, maia_launch_read_gaintrim, maia_setup (hymod gaintrim button events)
 - e. Fixed order (CSV table index order) for:
 - i. enable.ECH: maia_client_parameters, maia_launch_read_enable (effects order or returned ‘disable’)
 - 1. ‘disable’ effects: maia_update_parameters
 - ii. enable.EAN: maia_client_parameters
- e. Group table selection:
 - x. Group selection selected wrong detector

- xi. Need (*pm).groups.group[j].table in CSV index order
- xii. Read from Maia as: (*pm).groups.group[j].table[ref]
 - 13. 'ref' converts from detector to CSV index order
- xiii. Need to pass play in: maia_setup_groups_read, maia_launch_read_groups
- xiv. Does mean read/write group file routines store in CSV index order, which makes them transferrable from A, B, C

6. New version 6.6a

28 May 2013

7. Maia_launch

- a. Chart log-files
 - i. Now have a "logfile" entry in the Maia conf file.
 - ii. Can be different for multiple conf file set-ups
 - iii. Defaults to Maia_384A_log.csv in software home dir for AS.
 - iv. Added logfile to 'maia_defaults()', called from Maia_launch

8. Maia_setup

- a. Step-cal script builder 'build_kandinsky_script_step_cal'
 - i. Change to round low and move high to whole 3.0/1024 steps
 - ii. Modify Linearization script to adjust amplitude so that all of the peaks (count) are on whole values, to get an even step size between peaks.
- b. Save ECH disables to file
 - i. write_maia_enable:
 - 1. Somehow data array in write_maia_enable was type BYTE, which was interpreted as ASCII in str_tidy(). Forced int now.
- c. EAN selection gets lost when in Debug tab
 - i. Because tests of debug parameters from Maia (in 'maia-display' Notify code), cleared the EAN mode.
 - ii. Now, this does NOT happen if in Debug tab.
- d. Use 'config.dam.identity' for file-names (e.g. "384B3")
 - i. Except for layout maps (e.g. "Maia_384B-layout.png").
 - ii. All parameters file read/writes in Maia_setup.
- e. Apply Hymod and spec call setting:
 - i. Now write cals into spectra, groups, images from Apply Hymod code.
 - ii. Instead of via Notify, which was delayed often to after an update of *pm from background process, which trampled the new changes.
 - iii. "Get ALL cals" stills uses the notify mechanism, so the cal update call is still there in the Maia-Launch notify code for 'spectra-get-cals'.

9. Maia_update_parameters2:

- a. Default file names for E and T spectra structs, and file I/O
 - i. Set (*p).file to run number ((*pm).run.number)

10. Maia Rates

- a. Add Flux1 to rate window

11. GeoPIXE

- a. Image and Spectra windows
 - i. Reload plugins menu that reloads all existing and new plugins
- b. Add to header for Maia info
 - i. Use energy and title already in base device header.
 - ii. Decode {energy:19.5,sample:'string'} as 'energy' and 'title'
 - iii. Fill this in in get_header_info in Maia device
 - iv. In EVT, use new title info to set the 'comment' field

12. New version 6.5m

25 May 2013

13. Maia_setup

- a. Load parameters (see 'load-maia' event code)

- i. Use correct channel 'c' on chip, chip 'h' or detector 'd' for index 'q'
 - ii. (confusing having 'index' actually returned as detector number, change to 'dindex')
 - iii. for chip parameters set using first channel (c=0) on chip
 - iv. use (*play).ref[] to find index for selected detector numbers
- 14. Maia_update
 - a. ftp_connect
 - i. ftp_dir_list:
 - 1. for dates with time, but no year changed test for future months to look for day 't1' greater than present day 'tsys' PLUS 1.
- 15. New version 6.5l

23 April 2013

- 16. gaintrim plugin
 - a. Detect "/T" spectrum label to set .time.var extension
- 17. maia_setup
 - a. HYMOD panel, load Throttle default to .throttle.var files
 - b. 'enable-module-check' event:
 - i. set the enable bit in info strings to event.select, rather than the (*pm) enable bits, which have not been set here yet.
- 18. maia_launch_read_enable
 - a. update (*pm).throttle.on from *pdat
- 19. spectrum_index
 - a. 'mode' returns spectrum mode array (0="/E", 1="/X", 2="/Y", 3="/T")

19 April 2013

- 20. linearize_cuts_energies_spectrum_plugin
 - a. Reads cut labels in the form "Name 1.23" with energy
 - b. Does Linearization of this energy versus centroid of peak in each cut.
 - c. Needs to be calibrated for energy first.
- 21. Spectrum_Add
 - a. detects if charge = 0, then uses IC_count for weighting of DT
 - b. If both are zero, then uses equal weights
 - c. Fixed bug in summation of DT in singles_3 or array_4 mode
 - i. was $DT = DT * \text{charge} + \dots$

4 April 2013

- 22. maia_setup_apply_hymod
 - a. Now detects bad files for gaintrim, etc. and pops-up error.
- 23. maia_setup
 - a. Now HYMOD panel "Enable" checkboxes are labelled "Send" for linear, gaintrim, pileup, throttle, etc.
- 24. geopixe_environment:
 - a. Now does not pop up /tmp/geopixe error.
- 25. NSLS MCA device:
 - a.

21 February 2013

- 26. write_geopixe_image
 - a. Added /dump option to write simple binary dump for .dai and .xan data.
 - b. Added menu "Dump binary" to Image window.
 - c. GeoPIXE 6.5e.

18 February 2013

- 27. Processes – blog_client_*, maia_client_*

- a. clientname
 - i. Now open_socket accepts “client” and “retries”.
 - ii. These pass clientname from these processes to add these names to pop-up warnings.
 - iii. retries=0 used in these processes to have unlimited retries.
 - iv. But get error pop-up after 100.
- 28. open_socket
 - a. Now accepts “client” and “retries”.
 - b. These pass client to add these names to pop-up warnings.
- 29. socket_retry
 - a. Error pop-up after “retries” retries, or after 100 if retries=0.
- 30. GeoPIXE 6.5d

28 January 2013

- 31. Project and Group dir tags
 - a. blog_client_activity:
 - i. write new “name” to blogd
 - 1. combines machine_name with prefix string
 - ii. read one tag at a time (only wait for one):
 - 1. NO setgroup, setproject: in ‘sendprev’ mode to get current
 - 2. summary2, summary3: in ‘sendprev’ mode to get current
 - 3. activity: in ‘sendnext’ mode to wait and get next
 - iii. For group, project, build a stringify struct into *pb1
 - b. define
 - i. Add project to maia.run struct
 - ii. changed definition of maia_shared1 to accept number of detectors
 - c. maia_update_parameters2:
 - i. Decode (unstringify) *pb1 string into group, project
 - d. maia_launch:
 - i. Combine group, project into one status string.
 - e. Kandinski process “name”
 - i. Same trick as above for blog_client_activity applied to all blog clients, as well as DAQ clients.
- 32. GeoPIXE 6.5c

6 December 2012

- 33. Add XANES spectra sorting from dirs to Sort_EVT
 - a. EVT:
 - i. add xanes_dir=1 as another parameter to ‘p’ pars struct
 - ii. use “dir_mode” for sort_mode=1 and XANES_dir=1 for calls to get_header_info and flux_scan in evt_flux
 - iii. map scan details off in sort_mode=1 (XANES) mode.
 - b. evt_start:
 - i. veto extra (unnecessary) output_file processing at start
 - ii. use xanes_da_evt for XANES spectra
 - iii. add missing parameters, as in da_evt (devpars, etc.)
 - iv.
 - c. xanes_da_evt:
 - i. Use ‘k’ index to dirs as energy index (n_xanes in all).
 - ii. Build “images” of n_xanes * 1 “pixels”
 - iii. Build dead_fraction2, flux2 per dir, accumulate into dead_fraction, flux “images” across energies.
 - iv. Also build NN, NNPU, xanes_energy vectors
 - v. Accumulate “elapsed2” time by adding total dwell times for all files in dir.
 - vi. Read a new energy from DA matrix for each ‘first’ file in dir.

1. Only read whole DA once, then use pointers to extra energy versions for each dir
 - vii. Correct 'flux' for pileup losses and dead-time.
 - viii. If /flatten, then flatten to this flux.
 - ix. Output to a .xanes.csv file, with details.
 - d. get_header_info:
 - i. add /dir_mode, to take passed dir and read first file it finds.
 - ii. ****** This needs to be generalized ******
 - e. flux_scan:
 - i. add /dir_mode, to take passed dir and read first file it finds.
 - ii. ****** This needs to be generalized ******
 - f. extract_path:
 - i. make it able to process a vector of filenames.
34. GeoPIXE version 6.4x

16 November 2012

35. Move Kandinski parameter reads to background processes
- a. Use new call: "shared_memory_struct" to use a struct as a shared memory block
 - i. Much simpler than flat buffer form, although cannot have strings.
 - b. maia_client_parameters
 - i. read a lot of control and channel parameters into shared memory
 - c. maia_update_parameters, now copies from shared memory to *pm
 - i. (*pm).control = (*pshrmem).pdat.control
 - ii. (*pm).channel = (*pshrmem).pdat.channel
 - d. maia_client_parameters_slow
 - i. reads a lot of parameters for chart, extra to the above
 - ii. also copies throttle, rGamma, pileup, etc.
 - e. maia_update_parameters3
 - i. adds pars in shared memory to chart
 - ii. copies throttle, rGamma, pileup, etc. from shared memory
36. Speed now better, as long reads are in background processes
- a. Maia update now less than ~5%.
37. Build new as version 6.4v

15 November 2012

38. near_xy (image_routines)
- a. Make distance margin scale with 'zoom' passed as keyword, so that the accuracy does not get more demanding as the image is zoomed negative.
 - b. Used in "onbutton_image.pro"
39. Charge, flux in real-time DA images
- a. blog_client_da2
 - i. Set "Flux" element to match 'Flux0' or 'Flux1', if Maia h/w mode:
 - ii. Test maia_IC_name (from (*pl)[10: *]) for "Maia:scaler.FC0" form
 - iii. Test last word for FC0 or FC1
 - iv. If a h/w Maia counters, then set 'dflux' from 'Flux1' or 'Flux2'
 - b. image_flux_charge
 - i. returns a flag RT_mode, used in
 - ii. call this in image_details now to see updated flux, charge in RT mode
 - c. image_history_event
 - i. test RT_mode, and don't set 'charge' in RT mode when someone tries to edit flux, conv settings in the history window.
40. Offline Flux images
- a. Note that the sensitivity is not applied to the Flux0 and Flux1 images, only the Flux image, which gets used to flux, flatten and charge.
41. Traverse problems in analyze_image

- a. Add code to trim element list for line mode, like in area mode (see “n_el_none_blank”)
- b. Change code for xmdl, mdl to match length of conc, etc.
- 42. Show XY in Maia-control main window
 - a. Read “pixel[]” back in maia_update_parameters2
 - b. Add XY to legend in Maia_Launch_event, rates timer
- 43. Photons/sec
 - a. Added an estimate of ph/s from charge and total image dwell time (from dwell map) to the image history display.
- 44. Energy in var_val “scan.info”
 - a. Added code to read_maia_header_32 to decode struct {energy:12.345} in “scan.info” line in var_val tag 46.
 - b. Still also looks for energy PV in Monitor lists.
 - c. Later will also need code for extra fields in an expanded scan record
- 45. Maia.conf
 - a. Will now copy template.Maia.conf to home/.geopixe if no .Maia.conf file exists.
 - b. Change default path for Maia control files to .geopixe/maia in home.

21 October 2012

- 46. write_geopixe_image
 - a. Saving RT images from Maia_control uses the bounds to find the pixel size of the image. If the run is aborted (or low counts?) then the size will be smaller.
 - b. From Maia control the full size is stored in original_size, so we can compare the bounds to this and scale the mm size if needed.
- 47. Activity blog client
 - a. Collect FC0 from record and send back in shared memory, collected in maia_update_parameter2, and copy into (*pm).control.status.charge_rate.
 - b. Add this to rate window display.
- 48. Gaintrim plugin, ‘spectrum_index’
 - a. fixed the ‘spectrum_index’ routine to correctly use Linux filename that also contain “/” as used to delineate the “/E” tag.
- 49. Time-amplitude
 - a. Fixed an error in buffer size calculation to Long
- 50. Pileup, throttle
 - a. Change to “.pileup.var” and ‘.throttle.var’ file formats in varsh command form
 - b. get_pileup, get_throttle can read either form.
 - c. throttle3_select writes the new form
 - d. Change spectrum import, sort_evt reads to use these or ‘.txt’ file forms.

27 September 2012

- 1. Cater for multiple Maia and DAQ conf files (extension “.Maia.conf”, “.DAQ.conf”)
 - a. - GeoPIXE, Maia_Control, DAQ_Control v6.4o
 - b. Maia GeoPIXE conf parameters must be common to ALL Maia conf files
 - i. The same will be needed in DAQ device conf files
 - ii. new “name” parameter for building unique shared memory names
 - c. maia_launch
 - i. Combine device “name” with stub to make prefix for shared memory
 - ii. want these unique for each Maia running in parallel (and DAQs).
 - d. share_memory_buffers
 - i. Use prefix to make unique shared memory names
 - e. maia_defaults, daq_defaults
 - i. Add “/any” switch for use within device for simple common parameters.
 - ii. Add /reset to force the prompt for conf files for first call from launch.
 - f. spawn_blog_clients, spawn_daq_clients
 - i. Pass prefix and conf filename into routine and then to children as args
 - g. blog_clients (daq_clients)

- i. get prefix and conf-file from args
- ii. Use conf file in maia_defaults (daq_defaults) call
- iii. Use prefix to connect to correct shared memory areas

9 September 2012

- 1. Library
 - a. xsys_events
 - i. Fixed declaration for 'id' and 'type' to match 16-bit arguments
 - b. increment Lib version to 28
- 2. Maia
 - a. ScanInfo2
 - i. duplicated ScanInfo1 code
 - 1. blog_browse record, detail routines
 - 2. get_maia_details
 - ii. still need to detect units later
- 3. DAQ 32
 - a. Add first records to:
 - i. blog_browse: record, details
 - ii. read_maia

3 Aug 2012

- 1. read_old_da
 - a. fixed missing in struct: energy, thick, pmore
 - b. use get_lun, not unit=1
- 2. da_xstep_evt
 - a. needs to use matrix=matrix_file form
- 3. get_img
 - a. use device index and device name
 - b. use device_obj->big_endian() not old common block
- 4. get_csiro_dat, get_nac_spec
 - a. fixed device too
- 5. change to version 6.4h

5 June 2012

- 6. Added the following reform line to 'get_linearize', for version=3:
 - a. f = reform(f, nfrac, n)
- 7. Spectrum_select:
 - a. Added new entries to delete droplist for "not selected spectra", "not displayed spectra" and "spectra with invalid energy calibration".
 - b. OnButton_Delete_Spec_Select:
 - i. Use /Nots to delete not selected spectra.
 - c. OnButton_Delete_Active_Select
 - i. Use /Nots to delete not displayed spectra.
 - ii. Use /evalid to delete spectra with invalid energy calibration.
- 8. detector_select:
 - a. Add "get" button to set detector selection based on detectors found in a SPEC file.

25 May 2012

- 1. Define processed, valid, bad_xy and clipped in image struct as LL (Long64).
- 2. Change image read/.write to use this (version = -44).
- 3. Fixed absolute origin of image "Box" to account for offsets. Added "new origin" to history window.
- 4. Some notes added to "GeoPIXE internal workings.doc" on "absolute image coordinates".

23 May 2012

5. selective image region image sort:
 - a. define:
 - i. image:
 1. x_sub_range, y_sub_range added (no compressed values)
 2. sub_region flag added
 - ii. table:
 1. add xoffset, yoffset as long
 2. change original_sizes and nx,ny to long
 - b. read/write image files
 - i. fixed some errors with original_xsize (and y) when memory conserving extra compress applied.
 - ii. write/read sub_range, sub_region flag
 - iii. set sub_region if sub_range does not match original*compress
 - iv. write still PAINFULLY SLOW!
 - c. evt:
 - i. read template with sub_range, sub_region
 - ii. fall-back to test for
 - d. evt_start_image_increment2:
 - i. test (*pimg).sub_region and assign sizes of arrays to either:
 1. (*pimg).x_sub_region, ... or
 2. (*pimg).original_xsize, ...
 - e. da_evt, cut_evt:
 - i. save xoffset (and y) NOT compressed
 - f. image_routines: analyze
 - i. region include: offset
 - g. regions:
 - i. read/ write new pars: offset
 1. from version -24, change original_size and nx,ny to Long
 - h. image_table:
 - i. use offsets when extracting spectra → spec_evt
 - i. spec_evt:
 - i. get offsets from pmask for regions mode, set zero for total spectra
 - ii. added offset to read_buffer call
 - j. maia_device::read_buffer:
 - i. need to offset X1,Y1 for spectra return

Region transfer between images of different: compress and offset (not scale for now?)

Region loaded and passed to image window via 'image-region-select' notify. The original image will have different compress and offset to the new image, in general. Need to translate both (*p).pmark[0:1].x,y and (*p).q index array from old to new image coordinates. Note: don't change *p in situ, as it points to regions stored back in image. Need to translate only, into p2, and move results into (*pstate).pmark and (*pstate).q arrays.

19 May 2012

6. geopixe_update:
 - a. added proxy server support, v2.3
 - b. added "news.txt" file window, v2.4
7. selective image region image sort:
 - a. outstanding issues:
 - i. Regions do not store xoffset, yoffset yet.
 1. Can't use regions to/from windowed & normal images
 2. Will need new region file version to fix this.
 - ii. Spectra extraction from offset window image
 1. offset mask in Fortran?

- b. define:
 - i. image:
 - 1. x_sub_range, y_sub_range added (no compressed values)
 - 2. sub_region flag added
- c. evt_start:
 - i. trims EVT file list based on yoffset, yrange
 - ii. use “evt_start_image_increment2” now for cluster reconstruction
 - iii. set x,y range to sub-range if in sub-range mode and set offsets
- d. maia_device::trim_evt_files:
 - i. (if change this, then change in base_device too)
 - ii. pass yoffset, yrange, ycompress explicitly
 - iii. if no mask, then just use yoffset, yrange
 - iv. if mask, then clip to yoffset, yrange
 - 1. mask needs to use offset to EVT spectra
- e. evt_start_image_increment2:
 - i. make a new pro with “2” suffix in name, use this in evt_start
 - ii. it preserves the first strip image yoffset (and xoffset) now.
- f. da_evt:
 - i. pass xoffset, yoffset parameters and x_sub_range, y_sub_range
 - ii. set yoffset for each cluster node to max of this and node range.min
 - iii. adjust x,y origin/range for x,y offsets/range
 - iv. Note: xoffset, yoffset and xrange, yrange relate to full scan and are not compressed, while the xoffset3, xrange3, etc. are for window or stripe and are compressed.
 - v. stores compressed full scan size (xrange2) as ‘original’
 - vi. stores full scan size (micron) and origin (mm)
- g. others: follow changes in da_evt
 - i. cut_evt
 - ii. da_evt_xstep later?
 - iii. cut_evt_xstep later?
- h. evt:
 - i. “device” tab for device specific stuff
 - ii. add xoffset, yoffset to “scan” tab
 - iii. used x_sub_range, y_sub_range as well
 - 1. (or “X sub-range”, “Y sub-range”)
 - 2. else xsize, ysize become a problem
 - iv. New widgets:
 - 1. Mode: “Full Image”, “Select Image Window”
 - 2. on new map base:
 - a. “X offset”, “X sub-range”
 - b. “Y offset”, “Y sub-range”
 - v. From DAI template:
 - 1. sets x,y range by original size
 - 2. if offsets present, then set image_mode=1 and the sub-range to the actual image size
- i. set_image_minmax:
 - i. offset bounds (which are for full image) by offsets
- j. image_flux_charge:
 - i. changed line: “(*p).temp.charge_map)[(*p).bounds.xmin:(*p).bounds.xmax, (*p).bounds.ymin:(*p).bounds.ymax] = charge/float(max_pixels)” to offset cords by the offsets.
- k. bounds_mask:
 - i. offset bounds values

10 May 2012

- 1. parallel_config:

- a. add an error return (needs to be tested by cluster_client)
- 2. parallel_start:
 - a. added test for pshr ptr good
- 3. cluster_client:
 - a. added test for error from “parallel_Config”

9 May 2012

- 1. spectrum_add:
 - a. If input spectrum ‘data’ pointer was LONG then the cal-map did not work properly.
 - b. Forced spectrum to FLOAT.

7 May 2012

- 1. detector_geometry:
 - a. add ‘xtilt’ for global tilt about X axis.

2 May 2012

- 1. Maia device::
 - a. If a blog file is sorted that has no ET data, there seems to be missing pixels in the flux maps, etc. All PA words are processed, as are FC0, FC1, so this should not happen.
 - i. Found that test in Maia::read_buffer that sets max_y=ybase if no events.
 - ii. Changed this to max_y = (ybase > nsls_y) so that max_y tracks all PA y value changes.

20 Apr 2012

- 1. Revised and tightened string_packer routines, especially:
 - a. inumeric, fnumeric, gnumeric: rewritten (and added pnumeric)
 - i. stricter vailidity to match IDL now
 - b. float2, double2, longs2 (added fix2)
 - c. strip-trail_zero: re-written
 - i. Only strips after decimal point
 - ii. Uses long, float and double to get in standard form first
 - iii. accepts “places=” keyword
 - d. stringify: used strip_trail_zero() for data-types 4,5
 - i. changed places to -8, -16 for float, double
- 2. Revised str_tidy: using strip_trail_zero
 - a. accepts “places=”, “length=” keywords
 - b. Used now in 860 places across GeoPIXE/Maia in place of strtrim(string(),2)
- 3. File_Requester:
 - a. New options: /skip_if_exists, to not open if the filename passed exists and is readable on the path provided, and /skip_if_null, which means that a blank filename has been passed.
- 4. EVT button:
 - a. used file-requester /skip_if_exists, and /skip_if_null options on data files, linear, pileup, throttle, so only prompt for SPEC file name normally unless files not found.

19 Apr 2012

- 1. File_requester: added /skip_if_exists and /skip_if_null
 - a. /skip_if_exists – if file exists on the path specified, just return it
 - i. use in EVT region sort to avoid pop-ups
 - b. /skip_if_null – if file is blank then return blank
 - i. use in EVT region sort to avoid null linear, throttle, pileup pop-ups
- 2. options_popup:
 - a. added help_... options and the Help context-sensitive text field.
 - b. Updated align and shear plugins and “Generate DA matrix” code to use these.

18 Apr 2012

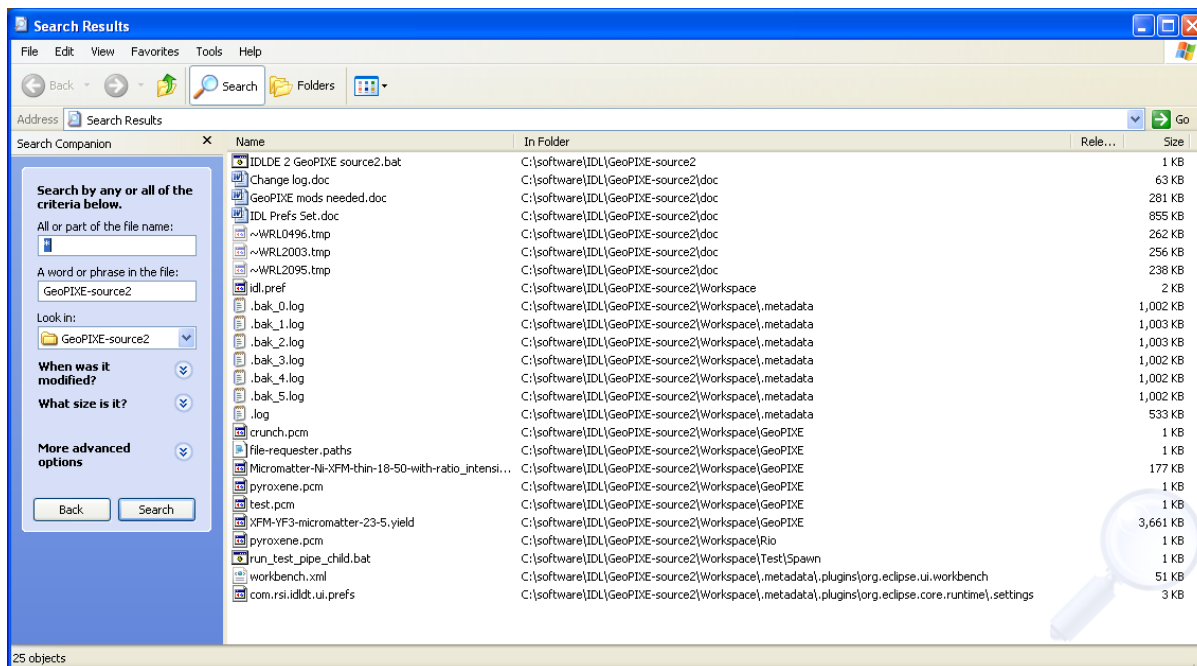
1. align_image – plugin to use Chris Jacobsen's img_align cross-correlation routine.
2. shear_correct_image – plugin to use above and Martin's idea of odd and even rows to detect and correct shear.
3. options_popup: finished implementation: defaults, help tracking

17 Apr 2012

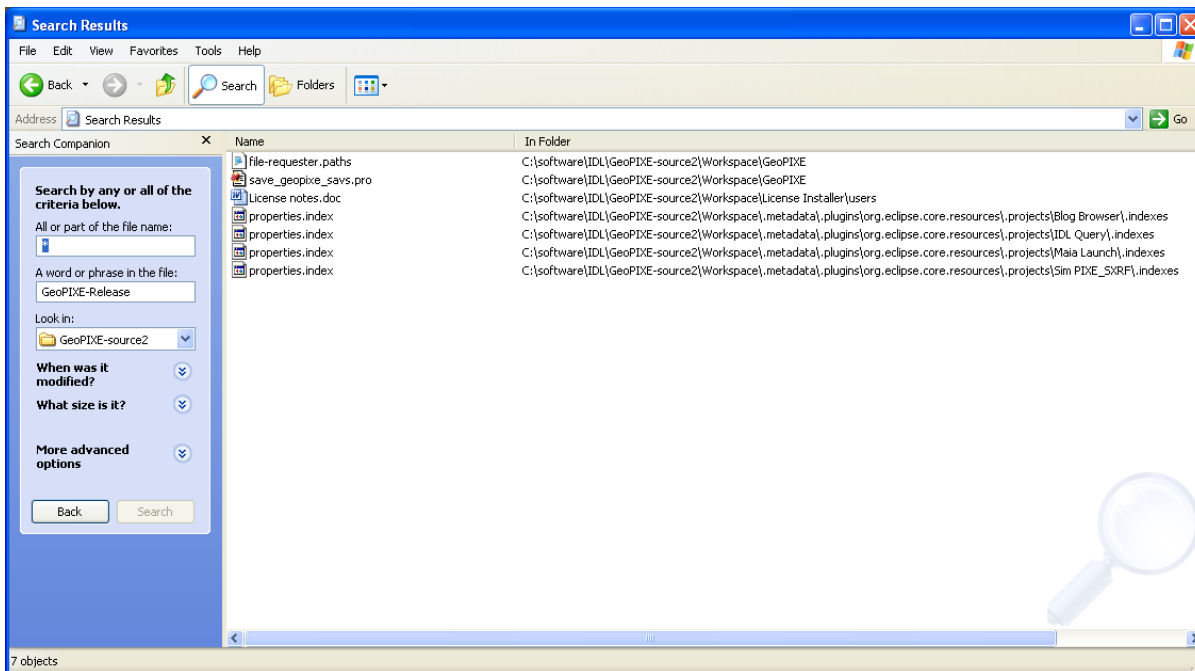
1. rebin error in cut_evt: must be integer multiple of dimensions.
2. Go through cut_evt against recently modified da_evt and update it.

15 Apr 2012

1. Copied GeoPIXE-source2 to GeoPIXE-source
 - a. Copied all files except the following:
 - i. IDLDE 2 GeoPIXE source2.bat
 - ii. Workspace\idl.pref
2. From now on:
 - a. **GeoPIXE-source: maintain only for bug fixes to version 6.3**
 - b. **GeoPIXE-source2: new development, starting with version 6.4**
3. Searched for all occurrences of "GeoPIXE-source2" to replace with "GeoPIXE-source":



4. Edited the following files:
 - a. idl.pref – copy from GeoPIXE-source2\Workspace and edit
 - b. IDLDE 7.1 GeoPIXE source.bat – was already OK.
 - c. save_geopixe_savs.pro – to copy SAV file to release\GeoPIXE
 - d. file-requester.paths – default one in GeoPIXE dir
 - e. workbench.xml – in Workspace\metadata\plugins\org.eclipse.ui.workbench
 - f. com.rsi.idltd.ui.prefs – in Workspace\metadata\plugins\org.eclipse.core.runtime.settings
5. Fix the following IDL project properties (to point to correct release dir). Later should change the properties probably:
 - a. blog browse
 - b. IDL query
 - c. Maia launch
 - d. Sim PIXE SXRF



14 Apr 2012

1. Cluster start bug
 - a. Array and individual modes – array dimension error
 - i. only for large meteorite 16278 – cores=4
 - b. YLUT ended in '0', which upset range method
 - i. Re-build and all is OK?
 - c. Fix DA_evt to use memory better – make image[] larger to start with and then copy attributes (flux) into this, rather than append to image.
 - d. Cut_evt: needs to be updated like DA_evt.
 - e. evt_start_image_increment: memory efficient?
 - i. `*(pimg).image = image` to copy new larger array back into `*pimg`
 - ii. should use `*(pimg).image = ptr_new(image/no_copy)`
 - iii. perhaps deallocate it first?
 - iv. Same for image_error, flux, raw_flux, dead_fraction, dwell, pileup.

12 Apr 2012

1. Fixed missing Zn excited by 3rd harmonic. Yield is zero, but an arbitrary limit is placed on “counts_per_ppm_uc” in “calc_da_matrix2” in the generation of matrix rows. Now the same limit is used on “matrix.yield” (which is also “counts_per_ppm_uc”) in “build_da_fit”. This restores Zn.
2. Overlays of DA fits on region spectra show an offset scatter peak fit in energy. This is because the original energy matrix is used, instead of the proper one. This is because the code in the spectrum_display notify “image-region-select” only tests the filename, not the energy. This has been fixed by loading the full matrix (all energies in the notify code, and then searching for the matching energy in “build_da_fit” before using the matrix.

10 Apr 2012

1. Wrote new function ‘options_popup’ as a general purpose input blocking widget with droplists, files, text strings and check-boxes, with map controlled by first droplist mask.
2. Extracted dir scanning code from ‘scan_batch_dir’ in ‘batch_sort’ as ‘scan_dir_evt’. It now builds and returns a struct array of details.
 - a. structs now contain energy – this can be used later to add energy column to ‘batch_sort’.
3. Modified ‘import_select’ to only show devices in /energies mode and return device index in struct.

8 Apr 2012

1. Fastcom_mp3_device__define changed for new ADC defaults from Sandia (Paolo). Init method common defaults change to 1,1 (was 0,2).
2. Compiled into local “interface”, but not copied to release or FTP site yet.

7 Apr 2012

1. New routine ‘spectrum_Import_energies’, off spectrum display Import menu, uses ‘scan_dir_evt’ and modified ‘import_select’ to select device and dir to scan for raw data and extract energies list, which is then saved as a .csv.
2. “Generate_DA_matrix” button in fit_setup now pops-up a popup (using ‘options_popup’) that asks for the type of DA matrix calculation to perform (defaults to normal DA) with option “XANES series DA matrix” to nominate a XANES energy list file and create a XANES series DA matrix file (“-xanes” appended to filename).
3. calc_da_merge_scatter:
 - a. Need to combine lines from “Compton” into “elastic” in “calc_da_loop”, if E_beam (first?) < 6.5 keV, only for series DA matrices both for first one.
 - b. Lots of variables in da_pars and peaks need to be combined:
 - i. peaks (to *pt): n_els, z, shell, n_lines, lines, e, intensity, yield, free, mu_zero, ratio_yield, ratio_intensity
 - ii. da_pars (make a copy too): A, mask, name, do_tail, counts_per_ppm_uc, cmdl
 - iii. also to A, mask, name, n_els derived from pars in calc_da_loop
 - iv. use ratio $A[\text{compton}]/A[\text{elastic}]$ to scale Compton rel-int before merging with elastic rel-int list
2. If read_da(energy=e) then pick up “scatter”, if read energy=0, then pick up “Compton” and “elastic”. This effects all use of “read_da”: append DA overlay for spectra on regions. If testing E_beam on read_DA, always look in series for a match in preference to the first matrix (which would not combine elastic and Compton).
- 3.

4 Apr 2012

2. Problem with Import versus Image versus Region spectrum average conc and flux.
 - a. DT became 0.5 (spec_evt) because total time (self.dwell_total) was zero passed from Maia→read_buffer for non-hardware flux cases.
3. Maia device – read_buffer
 - a. Fixed to accumulate self.dwell_total using elapsed time when no hardware flux counters used in buffers.
 - b. Make sure we use ms time units, like dwell returned from Fortran.
4. Flux_select:
 - a. Initial PV val in Import was zero (but showed as “1”)
 - b. Need to make initial value in *p par struct 1.0 to match “1” shown.
5. array_yield:
 - a. Sometimes (weak elements that reject all lines?) n_lines=0 for all elements and some arrays become zero length (e.g. cIntensity).
 - b. Change to limit length ‘nk1’ to a min of 1, when refit=0.
6. pixe_fit:
 - a. Change to limit length ‘nk1’ of cIntensity to a min of 1, when refit=1.

2 Apr 2012

1. Add spectra (re-map cal) bug:
 - a. regolith-82 spectrum has size not matching actual data
 - b. add tests in get_spec() to force consistency
 - c. change spectrum_add to use explicit size range in adds.

1 Apr 2012

1. PIXE_fit
 - a. **DANGER:** Added explicit tests on A5, A6 (tails) and A8, A9 (Compton). If they are negative after dA increment, then they get set back to previous iteration value.
 - b. Else, fit to Donut2x tails goes negative in amp (A5) in one iteration.

30 Mar 2012

1. Cal, Spectra notify:
 - a. Change action of “RA” button in Cal. Used to assume the use of markers and energies. Now if the A and/or B are changed instead, it uses these for a global cal change maintaining relativity between channels. Added new ‘cal-ab-RA-AB’ notify if A,B are changed.
2. append_DA_fit:
 - a. enabled backgnd
3. pixe:
 - a. fixed error in org offset for background (7) and pileup (4) parameters that stopped pure working for background enabled.

25 Mar 2012

1. Correct_element_lines:
 - a. ‘elastic’ peak added escape peaks also labelled with elastic index (as was ‘Compton’ before).
2. Extended DA matrix for XANES series energies
 - a. Just change elastic, Compton energy and loop in ‘calc_da_matrix’?
 - i. Need the energy list → shift scatter peaks
 - ii. peaks:
 1. e_beam – beam energy
 2. E[k,i] – energy of lines ‘k’, element ‘i’
 3. Lines[k,i] – line index for each line (test for scatter)
 - iii. calc_da_loop:
 1. repeat calc_da_matrix2 for each energy E
 2. shift all scatter peaks by $dE = E - E_{\text{beam}}$
 3. form array of pointers to DA matrix (E)
 4. each matrix DA needs E_beam added
 - iv. read/write/use DA matrix (w/ E added)
 1. read_DA() – returns DA matrix w/ pointers to series DA
 2. write_DA – write all DA in series
 3. additions to struct not an issue, but series is?
 - a. add to single struct return
 - i. E_Beam
 - ii. pointer to series pointer array
 4. free_DA: new free routine
 - a. use it everywhere new DA read_DA
 - b. DA_evt at end to free_DA (only if read_DA used to make a local copy after ‘matrix_file’ read)
 - c. evt_start: if read_DA then free_DA after
- b. evt_start
 - i. only pass ‘matrix_file’ now, not matrix anymore.
- c. da_evt:
 - i. use extended matrix (file or struct)
 - ii. get ‘beam_energy’ from header_info read in Maia setup in DA_evt
 1. look for match in DA
 2. assign ‘matrix’ to suit for use
 - iii. else just use simple top-level struct matrix
 - iv. need to deallocate extended DA matrix pointers
 - v. free_DA if opened locally.

22 Mar 2012

1. Add extra “Flux0”, “Flux1” planes to Maia device.
 - a. New method to maia: get_attributes() returns the names.
 - i. add dummy to base device.
 - b. read_buffer: accumulate FC1, FC2 if extra planes in flux
 1. How to avoid invalidating ALL device read_buffer routines?
 2. Use more planes for ‘flux’ array to pass extra data planes
 3. Detect # flux planes – extras are one less than this.
 4. Spectrum mode MUST keep flux as scalar only.
 - vi. Special ‘element’ planes:
 1. These new planes need to be added to ‘special_elements’ list?
 - a. else they get reported (e.g. regions) as sum divided by charge.
 - b. If use “Flux0”, “Flux1” for FC0, FC1, then this is done.
 - vii. image_increment – increment specified plane
 1. plane = i keyword
 - c. da_evt:
 - i. calls get_attributes and adds extra planes to flux if present.
 - ii. adds the extra ‘flux’ planes to image and attr names to ‘el’
 1. call get_attributes() to see if extras available for this device.
 2. allocate flux with extra planes
 3. after sort, add extra flux planes to image and el list
 - viii. image_correct:
 1. pass raw_flux too and add to pimg
 2. pass 3D flux to image_flux_flatten, plus raw
 - ix. image_flux_flatten:
 1. flatten elements to flux[:,*,0]
 2. flatten flux[:,*,0] too.
 3. flatten extra planes to raw
 - x. image format:
 1. define: add ‘n_attributes’ to image struct
 2. read/ write images – add n_attributes
 - xi. matching mods (as in da_evt) to:
 1. cut_evt, evt_start_image_increment, maia_launch

19 Mar 2012

1. pixe_initial – changed test for Cal significance (“wide_enough” line 241), to also allow wide_enough for eh-el>2.2 keV.
2. Went back to using all lines in tests for significant lines for Cal, FWHM and Tails: put tests for n_cal, n_fwhm, n_tail1, n_tail2 inside line loop on ‘i’.

24 Apr 2011

1. Base_device has methods: show_sort_options() and get_sort_ysize() for sizing/mapping of Sort options area.
2. Render Maia sort options in Maia_device object
3. Remove cluster from maia struct, add to (*p).
4. Read/write regions (device_name), version -20
5. Read/write spec (device_name), version -23

13 May 2011

1. Corrected “device eq 16” in spec_evt, evt_start, spectrum_preview including tests for pileup(), linear() methods, and deleted all ylut refs in spec_evt.

2. Corrected maia_ylut in evt_start_image_increment, added methods for get/write ylut to Maia and Base device. Made image_table EVT code use method, as well as write_ylut in evt_start_image_increment.

12 May 2011

1. Added missing xoffset, yoffset to read/write_geopixe_image as version -36.
2. Fixed errors in evt_start_image_increment.pro
3. Create test_evt_start_image_increment.pro to test 5496 stripe reconstruction.
4. Version -36 in read/write means that this cannot be retrofitted to old GeoPIXE 5.5
5. Move out local device code to separate projects, out of GeoPIXE src dir.
6. Make separate Builds for these (local files only).
7. Move all plugins into a “plugins” dir
8. Move all Maia blog client process SAV files into a “maia” dir.
9. Modify Image to look for Image plugins in “plugins”
10. Modify Spectrum to look for Spectrum plugins in “plugins”
11. Modify Fit_setup to look for Back plugins in “plugins”

11 May 2011

1. Added treatment of Y LUT within the Maia device object and the new method ‘trim_evt_files’.
2. Write new doc file “Object Oriented Device Driver Modules.doc”.
3. Move get_maia_ylut() call from EVT button code and into get_header_info() for Maia device.
4. Add output keyword to all get_header_info methods.

9 May 2011

1. Looked at all /get_lun to make sure that a matching close_file was always used, even after a file error to make sure that units are always freed.
2. Add devices: hasylab, sls.
3. Write notes on using Device Objects – GeoPIXE Internals.doc.

5 May 2011

1. Work through more device drivers: MPA3, APS, BD12, NSLS-MCA, Tohoku, Primecore, Zagreb, GSE-CARS.

26 Apr 2011

1. Delete XANES hack parameters from read_buffer (Maia first):
 - a. Leave commented out in Maia device::read_buffer/read_setup.
 - b. Also remove xanes switch

25 Apr 2011

1. Remove explicit Maia options from Image read/write – now done by object read/write_option method, which is called from read/write.
2. No change to read/write spec, they do not ref. Maia pars.
3. Now, remove refs. To maia parameters in read_buffer, all obvjcts, device_specific, da_evt, cut_evt, ...
4. Make Pileup, throttle, linear tests object-sensitive in da_evt, cut_evt, ...
5. Update da_xstep_evt, cut_xstep_evt (but it has ref. to “flux_ic”, but this par not passed yet – will bomb if xstep with IC data needed).

23 Apr 2011

1. Define_devices: use new objects
2. Device_list: use new objects and a fixed, offset list
3. Device_adc_offset: use **obj->start_adc()** method
4. Set_object_device: set a new device index and assign new obj, but don't use directly, use “device_object”, which returns the obj ref.

5. Device_object: sets a new device index and returns the obj ref. Use it to invoke methods:
(device_object(index))->method()
6. Device_object_name: returns device object name from index
7. Device_index_from_old_index: find new obj index from the old one, to use when reading DAI files, etc. to set the new index.
8. Changing default error return from get_header_info to 1 if no header available.
9. New version for write_geopix_image to -34, to change to writing device_name
10. New read_geopix_image:
 - a. Old version before 34: read old device index and convert to new index and device name using "device_index_from_old_index".
 - b. New version -34: reads device_name and finds new index.
11. Add new version code to write_geopix_image
12. New read_buffer, device_specific that call methods on object
13. Objects: mpsys, Maia done read_buffer, image_setup methods – it works w/ da_evt

22 Apr 2011

1. Set-up new GeoPIXE-source2 workspace, without explicit paths in prefs. Hence, needed to rename cw_bgroup to cw_bgroup2 and cw_fslider to cw_fslider2.
2. Still need to change Working dir in Prefs and restart (see "IDL Prefs Set.doc").
3. Started work on new Obj routines "open_device_objects.pro" to scan for device object routines (SAV and PRO, using "find_device_objects").
4. Started on base_device and Maia_device objects.