GeoPIXE Change Log

GeoPIXE-open-source

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

3 June, 2024

1. Maia device
    a. Added an options widget to adjust/correct a skew in X values, expressed as "correct X by one for every 'n' Y rows", with this set via a droplist.
    b. Read/write options now version -7 to cater for skew.x
        i. This will require this new version Maia device to read DAI file.
    c. 'Read_buffer' - corrects x1 by y1 (i.e. x1 = x1 - long(y1 / float(self.sort_options.skew.x)). This is done before any xoffset, so should work fine in cluster mode. Also done before any flip axis options, so applies to raw X1,Y1.
2. GUI Image Table XFM AS plugin
    a. image_table_xfm_as_gui_plugin
        i. Renamed this plugin from "image_table_xfm_as_gui_plugin" for the XFM beamline of the Australian Synchrotron.
        ii. Writes a CSV table with region details to be loaded into the online run spreadsheet.
3. GUI Image Table XFM NSLS plugin
    a. image_table_xfm_nsls_gui_plugin
        i. Adapted this plugin from the AS version for the XFM beamline of the NSLS-II synchrotron, BNL, NY.
        ii. Writes a CSV table with region details to be loaded into the online run spreadsheet.

31 May, 2024

1. Builder issue
    a. Some plugins are missing routines. These used to be done as IDLDE build options, not supported in new 'builder'.
    b. Needed to resolve missing routines and save/routines a new SAV file.
    c. Applies to: linearize2_spectrum_plugin, linearize_cuts_energies_spectrum_plugin, linearize_cuts_fit_offset_spectrum_plugin, linearize_cuts_spectrum_plugin, linearize_spectrum_plugin.
    d. Need to add: *lmfit*.
    e. Add a new place for miscellaneous resolves: 'resolve_misc'.
2. Resolve_misc:
    a. Resolve routines: *lmfit*.
    b. Users who add plugins can also resolve their own added routines before saving a SAV file. But many basic ones can be added here, as this is not supported yet in builder.

19 May, 2024

1. Notes about /modal, /floating, blocking
    a. Originally, 'file_requester' was setup as a normal modal widget, which blocks until is closes. It's also floating, which means it sits in front of other windows.
    b. Blocking is similar, keeping control in xmanager near the end of the main file requester program. We don't want to set no_block=1 on xmanager. However, we should <u>never</u> set no_block=0, which does some weird stuff, like disable debugging. Simply, *omit* the no_block keyword to xmanager to achieve the same result. See 'file_requester code.
    c. Then file_requester was changed to permit a progress bar popup while searching for a file match using "Find". i.e. It was found that you could use modal=0, floating=1 on the top base widget and *omit* no_block on xmanager.
    d. However, then file_requester could not work properly if called from another modal widget (e.g. 'flux_select'). In that case, we use another option '/within_modal' to remind

file_requester to use /modal (and disable the Find). Only use '/within_modal' when called from a modal widget.

2. File_requester
    a. New '/within_modal' option, if called from a modal popup (e.g. 'flux_select').
    b. This means that the file "Find" does not work as the Progress window cannot function properly then.
3. Flux_select
    a. As this is a modal popup, all calls from it to 'file_requester' use /within_modal.
4. Spectrum_load_prep
    a. Call 'file_requester' for pileup, linearize and throttle with 'updir=3' so that it looks in the local neighbourhood for a file match.
5. Version 8.8m.

15 May, 2024

1. Plugin: Cal by centroids
    a. peak_cal_cut_centroid_spectrum_plugin
        i. Use a common to save the 2 energies, so they appear next invocation.
2. Write_source, write_pink, read_source, read_pink
    a. Only set 'file' in struct if file opened here, not if 'unit' passed.
3. Fit_results/save_fit_resilts
    a. Write a new version -16 and write 'continuum' first, then 'model'
    b. Then call either source or pink write.
4. Read_fit_results
    a. New code for version -16, to read 'continuum' first, then 'model'.
    b. Then call either source or pink read.
5. Pink_calculate
    a. Use file_requester(/skip_if_exists) to search local tree and also along 'path' tree for a file match.

6 May, 2024

1. Continuum source
    a. Moved back into 'main/xray' and 'main/' for 'source_setup'.
    b. Delete "continuum_Source" plugin project.
2. Pink beam source
    a. New pink beam routines in 'main/xray' and 'main/' for 'pink_setup'.
3. Pink_setup
    a. New setup for a pink beam that parallel 'source_setup' for lab sources.
    b. Uses FE beam spectrum and up to 4 mirrors, based on CXRO reflectivity data files.
    c. Disable inline optics extras for now.
4. Define
    a. New /pink for pink beam
    b. Set model=1 for lab, model=2 for pink, as was not used before.
5. Layer_setup
    a. No source plugin load, fixed new droplist options for photon sources.
    b. (*p).source is now a pointer to either a lab source or pink beam 'beam' struct
    c. (*p).source → *(*p).source in most places.
    d. Cater for 'new-pink' Notify event, similar to new-source.
    e. 'beam-mode' now just handles lab and pink in "else".
    f. 'load-source' now handles cases of 6:lab, 7:pink beam.
    g. 'new-source' launches 'source_setup' or 'pink_setup' according to beam.mode.
    h. Add 'ppink' to pstate for notify to pink_setup.
6. Geo_yield2
    a. At the end, add "rel_int[1:*,q] = 0.0" for the zero 'q' array, to avoid NaN 'rel_int' for elements where the yield drops to zero.
    b. Some bad elements get through, despite 'zero', so add this to 'q' test:

        i. Q = where((zero eq 1) or (finite(rel_int[1,*]) eq 0))
    c. Catch all /0 conditions that affect: branch_ratio, ppm, mass_yield, total_yield, rel_int.
7. Geo_array_yield
    a. Catch all /0 conditions that affect: rIntensity, enhance, rY.
8. Select_element_lines
    a. Detect lack of "Lines" tag in 'beam' struct and set add_lines=0
9. Write_yield
    a. Write beam.model before beam, if 'use_beam'.
    b. New version -11
10. Read_yield
    a. If version -11, after read 'use_beam', read 'model'.
    b. If 'model = 2 then switch to a pink beam 'beam' struct.
11. Read_beam
    a. Read a text file of energy versus beam or CXRO reflectivity data.
    b. Has a 'skip=' option to skip header of CXRO files.
    c. Has option 'remap_energy=' to remap E,data onto a given energy vector.
    d. Clip output to >0
12. Pink_calculate
    a. Just use FE beam * mirror and filter reflecitity/transmission, with no solid angle, etc. terms applied. Commented out for now.
    b. Disable inline optics now, as this involves solid-angle considerations also.
    c. Set e_beam, which is used to 'init_xrf' in 'get_lines':
        i. pink.energy = E[max(where(spec gt 1.0e-10))]
    d. Add pressure and temp args to 'transmit' like in 'source_tube_spectrum' called from 'source_calculate' to cater for spectra with P.T values.
13. Read_pink
    a. Read a pink beam file, like read_source.
14. Write_pink
    a. Write a pink beam file, like write_source.
15. fit_recalculate_yields
    a. Add case to either 'source_calculate' or 'pink_calculate' based on beam.model
    b. Note for non continuum beam these return passively.
16. Version 8.8k.

22 Apr, 2024

1. Maia Launch
    a. Maia_launch_version2
        i. Cater for 'number' zero for spectra, et2d, roi.
        ii. Zero means the variable vanishes in Kandinski.
    b. Maia_launch_initial, maia_launch_read_enable, maia_launch_read_groups
        i. Only set spectra parameters if number not zero
    c. Maia_launch
        i. Only setup groups shared memory is number > 0
        ii. Catch anywhere shrmem referenced
    d. Maia_update_group_spectra
        i. Return if no shared mem
2. Spawn_blog_parameters
    a. Pass in region enable, else veto spawn of region spectra
3. Maia_setup
    a. Maia_setup_event, maia_setup_apply_groups
        i. Only set spectra parameters if number not zero
4. Maia_client_parameters
    a. Test for (*pm).number.spectra, roi and skip enable.
5. Startupp
    a. Construct path to 'maia' and 'daq subdirs correctly.

6. pixe_fit
    a. Set default value for 'sum_deficit' to 0.5%
7. Version 8.8j

8 Apr, 2024

1. File-requester
    a. File_requester_preview
        i. Use *(*(*pstate).p).pfile (array) for files, rather than (*pstate).file (concatenated list).

14 Feb, 2024

1. Blog browse
    a. Change format of output to correct a field width.
2. GitHub
    a. Fresh load of GeoPIXE to local Git and to GeoPIXE on GitHub at URL:
    b. https://github.com/CSIRO-GeoscienceAnalytics/GeoPIXE as branch "main".

27 Nov, 2023

1. Use *WinMerge* to view and merge changes across from GeoPIXE-source3 to *Open Source* (between Mar 27 and Nov 27). Take care to:
    a. Not merge across differences due to licensing.
    b. Watch for different path constructs, e.g. for Wizards.
2. Need to run all tests

24 Nov, 2023

1. GeoPIXE
    a. gImage
        i. Add menu for Export → Simple Image -> Save as TIFF
        ii. Call Image_Save_GIF, Event, /TIFF
    b. Image_Save_GIF
        i. Add /TIFF option
        ii. For TIFF use 'write_tiff' to save as 32-bit float TIFF.

8 May, 2023

1. GeoPIXE
    a. Removed modules with non CSIRO code, although the code had been supplied by the laboratories to enable an import object or plugin to be devised for their data and attribution was included.
        i. These include:
            1. Devices:
                a. "APS LST", "ESRF EDF", "GSE-CARS MCA", "Lund VME", "NSLS MCA", "NSLS NetCDF".
            2. Plugins:
                a. "Image Align", "Image Combine Align", "Image Shear Correct".
            3. Background:
                a. Bayes.
            4. These may be added to the GitHub repository by the labs concerned at a later time.
1. Builder
    a. Add build for GUI plugins.
2. Eclpse "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

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 catagories ('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. Version 8.7o.