GeoPIXE Change Log

GeoPIXE-open-source

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

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.