Getting started with GeoPIXE

Contents

GeoPIXE from GitHub

The archive on GitHub contains all source files and essential data and documentation. It does NOT include metadata (Workspace/.metadata directory) for the *IDL Development Environment* (IDLDE) based on Eclipse. What is missing are Eclipse project definitions and the build settings for each project. Import of the projects is easy (see below and the documentation "**doc/GeoPIXE Software Organization - open.pdf**" for details). But we must set a couple of settings regarding management of the !path in IDL. These are detailed below.

Building can be easily handled using "**Builder.pro**" (see below). However, you can setup Eclipse IDL build settings if you like, as described in section "Eclipse environment and organization".

Downloading

If you are reading this, then perhaps this is done for the software… But for worked example data, you need to also download the Demo data, which is archived in the CSIRO DAP at DOI: https://doi.org/10.25919/mc2f-1979

Install IDL

Running GeoPIXE requires at least IDL runtime support. With IDL installed, but not licensed, you can run GeoPIXE using IDL *Virtual Machine* runtime support. To program in IDL and build GeoPIXE source, you will need an IDL license.

Running GeoPIXE from the compiled 'geopixe' directory

Once IDL is installed, you can run GeoPIXE simply by double clicking on "GeoPIXE.sav" in the "geopixe" folder within the "workspace" tree. It will note the absence of a "geopixe.conf" file, but will then create one for you in your <home>/.geopixe directory.

It helps to have a working understanding of the *Fundamental Parameter* approach used for quantitative analysis in the workflow through GeoPIXE, and some experience with the Demo data. See the ***GeoPIXE Users Guide*** and the ***GeoPIXE Worked Examples*** PDF for worked examples with step-by-step tips. Both provide examples of the main tasks of fitting spectra to generate the ***Dynamic Analysis*** (DA) image projection matrix; using this DA matrix to process full-spectral data to deconvolute elemental components and project separated elemental images; and exploration and processing of the images to first verify their accuracy, including extracting spectra from observed features (region shapes or element-element Associations), and make corrections and then to explore their content.

If you want to compile GeoPIXE code in IDL, then you'll need an IDL license and continue setting up the Eclipse IDLDE environment.

Setting up Eclipse IDLDE environment

To import all projects, use the "*File->Import->General->Projects from folder or archive*" menu and navigate to "Workspace" in the local downloaded GeoPIXE "Workspace" directory as the "Import source" folder. Then select all project folders and "Finish". This will import all projects. See the documentation "doc/GeoPIXE Software Organization - open.pdf" and section "Eclipse environment and organization" for more details of the Eclipse environment.

However, this does not import project settings for building via the IDLDE menu. Building can be handled using the "builder", as outlined below (see section "Building GeoPIXE"). But you need all projects imported (see above) and there are some settings that must be set now. For just the "geopixe" and "Fortran" projects, right click and select "Properties" for each. For the "IDL project properties" group, uncheck the option "update IDL path when project is opened or closed", so these projects are not added to the path. You can also do that for the "Default" project, if you are not using that.

See the documentation "doc/GeoPIXE Software Organization - open.pdf" for details of the "***Eclipse environment and organization***". Building GeoPIXE is covered in another section "***Building GeoPIXE***".

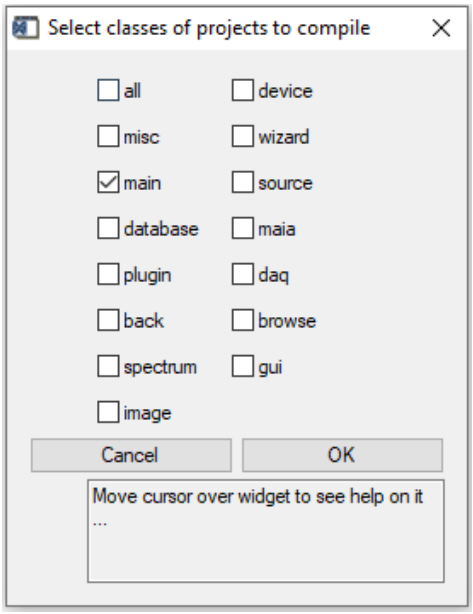
Building GeoPIXE

GeoPIXE and its modules can be built using IDLDE menus or using the "Builder" PRO script. The former requires the "properties" of every project to be setup. The *Builder* approach avoids this. But does require a two-stage approach: (i) run *Builder* (double click on "builder.sav") to construct a script ("build.spro") for building the selected (or all) components, and (ii) run the created "build.spro" at the **IDL command line**. This is described below.

NOTE: The file "build.spro" will end up in the project "main" directory, if builder is run within IDLDE and this is set as the working directory, but may end up in the "geopixe" directory if run from the builder SAV file at runtime.

Compilation of GeoPIXE main using the Builder PRO

In the absence of IDLDE properties being setup, the simplest way to perform a full or partial build is to use the "builder" (builder.sav). This is the assumed approach for source downloaded from GitHub. Builder by default pops up a list of groups of projects to build. Select "all" to build everything. To build just the "main" dir tree of GeoPIXE program PRO and library routines, select "main" only. The result of running Builder is a script "build.spro", which needs to be executed at the **IDL command line** ("@build.spro"). Use "cd" to select the correct path first, to where build.spro is located. A pop-up will provide hints to this, including the paths.



Compilation of modules

Each project module for GeoPIXE can be built either using IDLDE and the “build” menu for the project (e.g. “main” above) or by using the “builder” PRO.

The result of the GeoPIXE “main” compilation (see above) is the GeoPIXE.sav file, which does not include those modules loaded at run-time (e.g. devices, plugins and wizards, “geopixe_parallel.sav”). These need to be compiled separately, by building the corresponding IDL projects. Similarly, each device object project, wizard or image/spectrum plugin must be built individually and not include main GeoPIXE routines.

The IDLDE build properties for plugins, wizards, devices and the Maia and DAQ modules do not do a resolve_all, and place the compiled SAV files in the “geopixe” runtime directory tree.

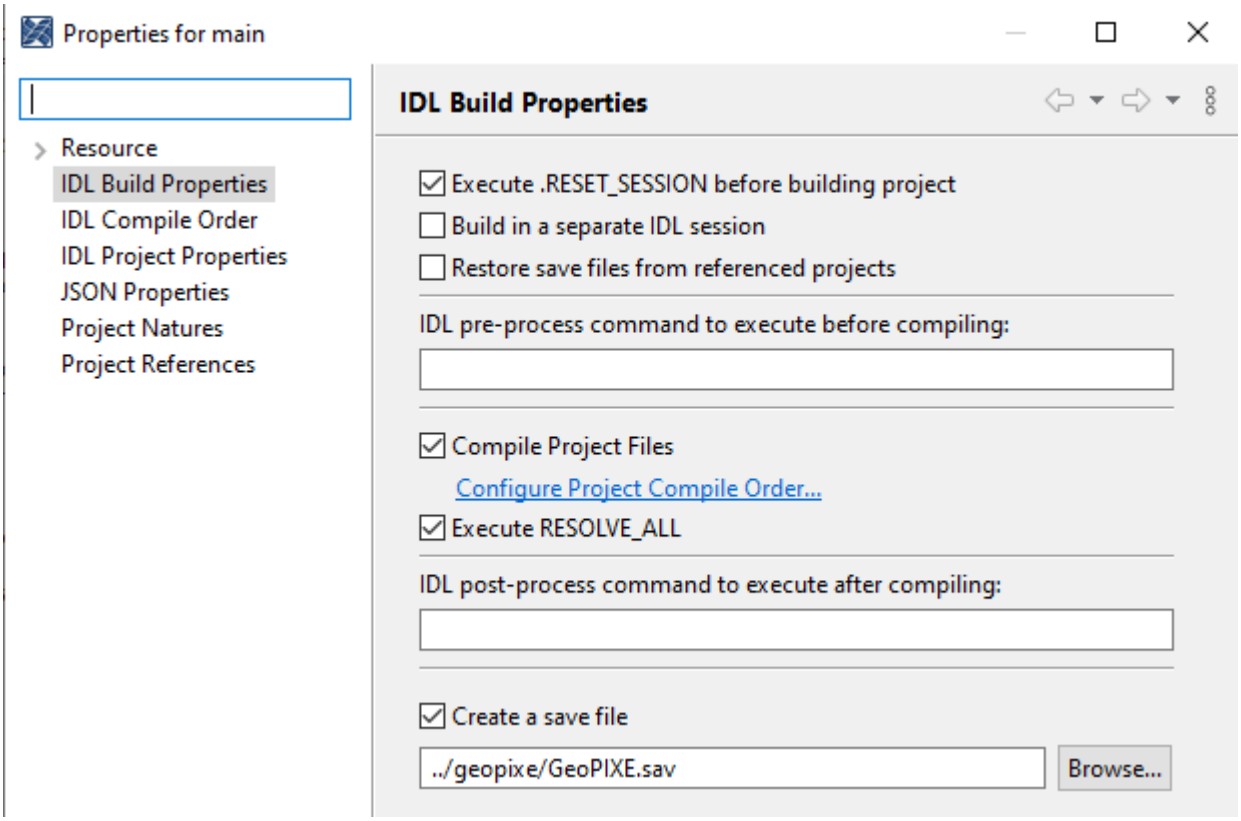
In the absence of properties being setup for each class of project, the simplest way to perform a full or partial build is to use the “builder” (or its SAV file in runtime). Builder by default pops up a list of classes of project to build. Select “all” to build everything. To build just one class of project module (e.g. plugins) select the class from the pop-up list. The result of running Builder is a script “build.spro”, which needs to be executed at the **IDL command line**. Use “cd” to elect the correct path first, where build.spro is located.

Compilation of GeoPIXE main using Build menu

For the IDLDE Build menu approach, the properties described above for the main GeoPIXE project “main” outline how GeoPIXE is compiled. In detail, this involves:

1. Reset of the session, so no compiled routines are in memory.
2. Compiles all routines in the GeoPIXE directory.
3. Executes “resolve_all” to resolve all references anywhere on the IDL search Path, which is across all open projects and IDL libraries.
4. Writes the compiled “GeoPIXE.sav” into the runtime “geopixe” directory.

This can be done in IDLDE by selecting the menu item “Build” for the project “main”, if the properties are setup correctly (see above). Right click on project “main” and select “Properties” to setup the build properties. The result of the GeoPIXE compilation is the “GeoPIXE.sav” file, which contains all referenced procedures and functions, but not those loaded at run-time (e.g. devices, plugins and wizards, which are not referenced yet, as well as “geopixe_parallel.sav” – see below).



See the documentation "**doc/GeoPIXE Software Organization - open.pdf**" for details of compiling other projects in GeoPIXE.