DUNE Xerosere with Phlex

Data, algs, files, config, units, devel, build.

Brett Viren

September 24, 2026

Xerosere

1. *Plant evolution from arid, water-limited ecosystem to a lush verdant forest.*
2. *Development of a DUNE software ecosystem on top of the Phlex backbone.*

Topics

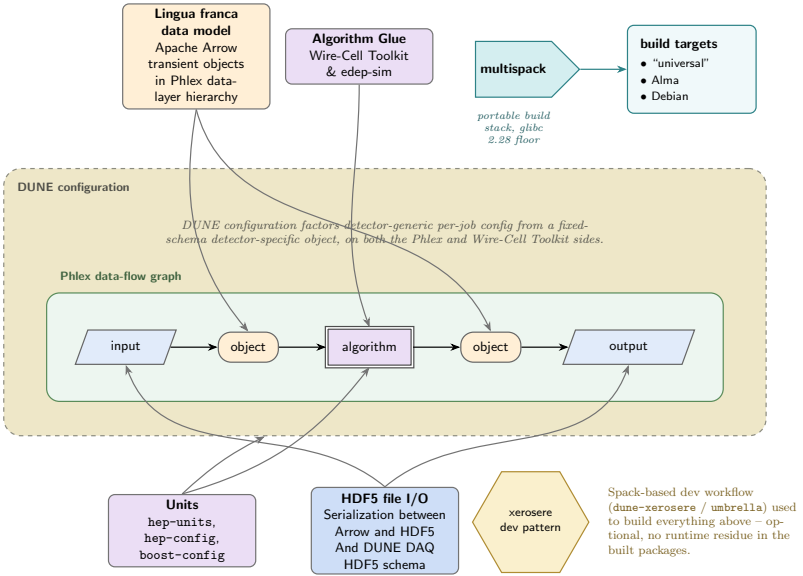
Where we are - various, DUNE processing chains in Phlex with many bells and whistles

- **DUNE algorithms** glued into Phlex
- **DUNE configuration** for Phlex and Wire-Cell Toolkit
- **DUNE units** via hep-units, hep-config, boost-config.
- **DUNE *lingua franca* data model** based on Arrow.
- **DUNE DAQ HDF5 codec** and **Arrow/HDF5 serialization**.
- **Xerosere development environment** patterns and helper scripts
- **Universal-binary Spack-based build method**.

Where to go next

- **Bringing in others**.

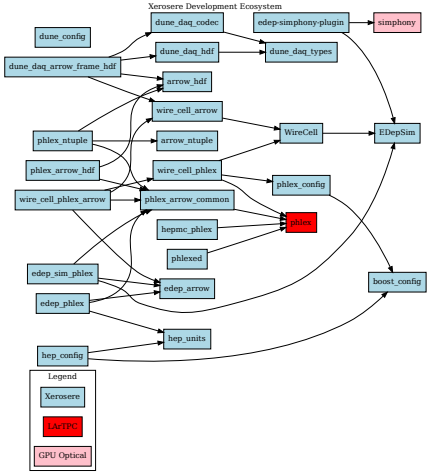
Xerosere conceptual architecture design



Current Xerosere package dependency graph

The ecosystem is becoming verdant and real.

- Existing packages that run DUNE software on Phlex ⟶.
- Well-factored, modular packages with careful dependency management, avoiding lock-in.
- Integrations: Wire-Cell Toolkit, edep-sim/Geant4, DAQ data, HDF5, Arrow, units.
- Bonus, BNL's **GPU-accelerated plugins** work:
  - WCT / **SPNG** - Signal-Processing, Next Generation
  - edep-sim / **Simphony** - advanced version of Opticks



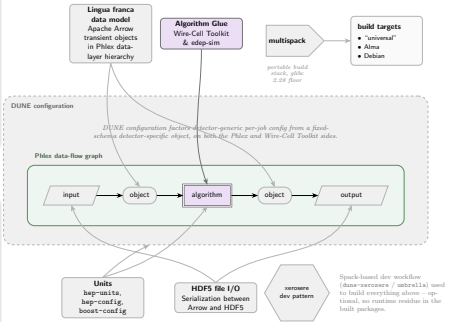
DUNE algorithms in Phlex

Venerable work horses

- **`wire-cell-phlex`** support for popular Wire-Cell Toolkit jobs (sim, sigproc, 3D imaging, …)
- **`edep-sim-phlex`** run edep-sim / Geant4 in Phlex (robust against Phlex/TBB threads!).

Newly written

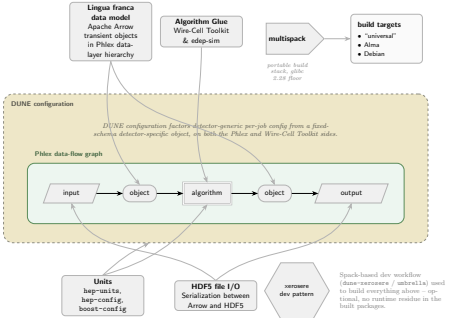
- **`dune-daq-codecs`** DUNE DAQ raw data encoder/decoder and online/offline channel maps.
- **`hepmc-phlex`** HepMC3-based kinematics
- **`edep-phlex`** ionization depo generators



DUNE configuration

- **`dune-config`** covers Phlex with Wire-Cell Toolkit, edep-sim, DUNE DAQ file reading.
  - More can be added, of course.
- Well-factored configuration based on `dunereco`.
  - **detector independent** job configurations.
  - parameterized by **detector-specific** objects.
- Supported detectors so far: pdhd, pdvd and pdsp.

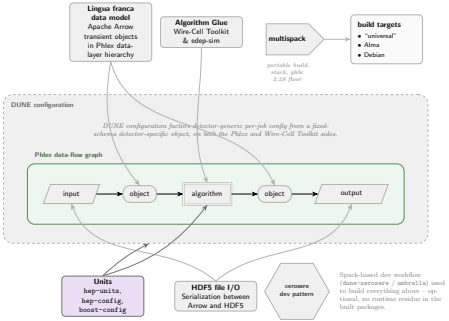
A separate `phlexed` package provides a `phlex`-like main program that adds proper Jsonnet CLI support.



DUNE Units

- **`hep-units`** CLHEP system of units in extensible form with code generator for C++, Python and Jsonnet (and thus JSON, YAML).
- **`boost-config`** General C++ header-only helpers to convert between C++ `struct` and `boost::json` objects with minimal or no code instrumentation.
- **`hep-config`** Glue between the two to allow automatic parsing of units-as-strings for languages that lack variables and arithmetic (JSON, YAML, FHiCL).

Details DUNE S/W Arch. 2026-08-21
https://indico.fnal.gov/event/75778/contributions/347641/attachments/200812/280154/hep-units.pdf



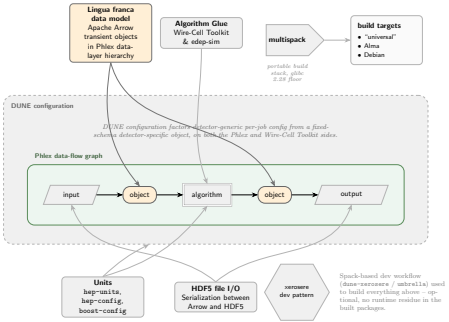
DUNE *lingua franca* data model

Previously[a] introduced *lingua franca* data model idea.

- Factor data types into a base **format** and higher level **schema** forms.
- Helps break $N \times M \to N + M$ complexity.
- File I/O need only know about **base format**.

First considered DIY but then discovered **Apache Arrow**

- Provides **format** and **schema** and solve other problems.
- Well matched to HDF5 and RNTuple file-level formats.



[a] DUNE collab 2026-05-22 https://indico.fnal.gov/event/71671/contributions/341334/attachments/198068/275613/datamodel.pdf

Integration-related Arrow packages

Data packages

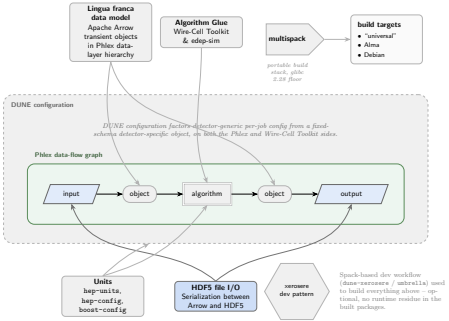
- **`arrow-ntuple`** General, simple and light-weight N-tuples.
- **`phlex-ntuple`** Phlex glue for arrow-ntuple
- **`wire-cell-arrow`** `Depo`, `Frame`, `TrackSegment`, `Tensor`, and their Sets.
- **`edep-arrow`** `Photons` and `Segments` (more truth to come)

Converter packages

- **`phlex-arrow-common`** Phlex data-layer hierarchy helpers
- **`wire-cell-phlex-arrow`** WCT `IData` and Arrow converters.
- **`edep-sim-phlex`** edep-sim data converters (and edep-sim integration).

DUNE DAQ HDF5 and Arrow/HDF5

- **`dune-daq-hdf`** DUNE DAQ specific file-level schema handling.
- **`dune-daq-codec`** DUNE DAQ codecs (inc. online/offline channel map).
- **`dune-daq-types`** replicate DUNE DAQ "dataformats".
- **`arrow-hdf`** generic serialization between Arrow and HDF5
- **`phlex-arrow-hdf`** Phlex glue for HDF5 file I/O

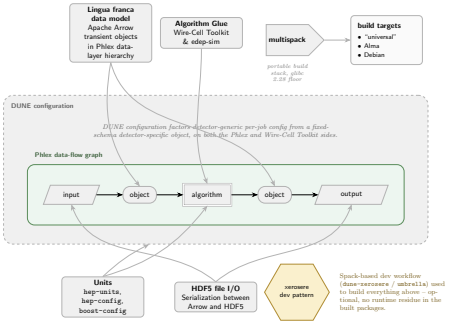


Xerosere development environment pattern

Development environment using Spack and a simple file tree convention.

- Develop multiple packages against multiple Spack environments.
- Develop multiple versions of the same package.
- Keep all build and install areas distinct.
- Manage the different shell runtime environments.
- Devel products are not dependent on xerosere.

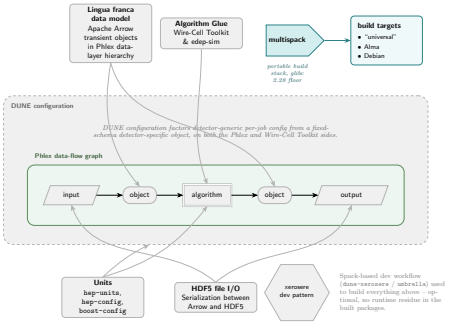
Since introduced (Phlex AWG 2026-07-13), I extracted the original umbrella shell script from `dune-xerosere` to a stand-alone xerosere Python package/CLI.



Multispack provides a build environment making generic binaries

- Glibc promise: build on old glibc, run on new glibc.
- This is how **manylinux** populates the Python ecosystem.
- Build inside host-independent containers for hermetic, repeatable and documented environments.
- Deployment options: Spack install in `/cvmfs`, Spack binary buildcache or Conda/Pixi channel via **spaxi**.

https://github.com/brettviren/multispack
https://github.com/brettviren/spaxi



Bringing in others

Xerosere is…

- … a good first step but DUNE still has a lot of work to do.
- … missing help/docs/workshops to help bring others up to steam.
- … ready for “motivated self-learners” to use and extend.

Let’s talk if you want to get involved and how best to help others.

Foundations

To let a thousand flowers bloom efficiently, DUNE must “standardize” on some basic parts:

- Pervasive, global use of one **system of units**.
- One *lingua franca* data model (and a several package-specific ones).
- Shared **configuration patterns** with composable, detector-independent job config parameterized by detector objects.

Summary

- Xerosere provides **today** a full-chain for DUNE running in Phlex that covers from raw-data to particle tracking and detector response simulations and signal processing.
- The Wire-Cell Toolkit and edep-sim libraries have been **integrated** into Phlex.
- A DUNE *lingua franca* data **model based** on Apache Arrow is now available.
- Phlex-based **HDF5 file I/O** between Arrow and DUNE DAQ schema has been developed.
- Comprehensive support for **system-of-units** and **configuration** exists.
- The **xerosere development pattern** has proven to be empowering, flexible and generic and is now available for others outside of the `dune-xerosere` package.
- The **multispack** universal binary build method has been proposed, developed and tested.

FIN

Backups

Multispack update - picking the glibc version

- Original choice was glibc 2.28 as provided by Alma 8 and manylinux.
- Later I found that Alma 9's glibc 2.34 is not so new that it rules out many others distros.
- Glibc 2.34 binaries will run on Debian 12+, Alma 9+, SLES 15sp6+,

So, we can gain multispack benefits, keeping essentially the same building on Alma 9.

- Main thing is the bootstrap ladder to give a pure GCC + gcc-runtime builder.

glibc

distro	glibc	gcc	family
AlmaLinux 8	2.28	8.5	rpm
AlmaLinux 9	2.34	11.5	rpm
AlmaLinux 10	2.39	14.2	rpm
Rocky	2.28/2.34	8.5/11.5	rpm
RHEL	2.28/2.34/2.39	8.5/11.5/14.2	rpm
Fedora	2.39/2.40/2.41	14/14/15	rpm
Debian 11	2.31	10	deb
Debian 12	2.36	12	deb
Debian 13	2.41	14	deb
Ubuntu	2.31/2.35/2.39	9/11/13	deb
SLES	2.31/2.38/2.38	7.5	rpm(zypper)
openSUSE Leap	2.31/2.38	7.5	rpm(zypper)
openSUSE TW	2.42	15	rpm(zypper)

Odds and ends

`edep-simphony-plugin`

Developed by Xuyang Ning, adds Simphony's advanced GPU acceleration of photon propagation to edep-sim.

`phlex-config`

Header-only C++ schema structs based on `boost-config` to give common code-paths to conversion between `boost::json` and C++ structs.

`phlexed`

- **`phlexed`** command adds proper Jsonnet handling to `phlex`.
- **`graphlexed`** renders Phlex data-flow programming graph to GraphViz dot.

