Spack-based build of edep-sim and eic-opticks

And how to use it for development

Brett Viren

March 3, 2026

Topicks (sic)

- Building the packages with Spack
- Building a developer environment
- Plan for initial development

Base spack build

Spack

```
git clone --branch v1.1.1 --depth=2 https://github.com/spack/spack.git
source spack/share/spack/setup-env.sh
spack compiler find
spack bootstrap now
```

Our packages

```
git clone https://github.com/WireCell/edep-opticks-spack.git
spack repo add edep-opticks-spack/edep_opticks/
spack install --jobs 16 edepsim+opticks cxxstd=17
```

Comments

- In principle, `cxxstd=20` can be used instead (need C++20 in FNAL ecosystem).
- This is purely a **local build**, no dependencies on WCWC, UPS, CVMFS, FNAL-spack, etc.
- The `+opticks` variant causes `eic-opticks` to build though edep-sim does not yet actually depend on it.

Spack View based developer environment

```
mkdir hack && cd hack/

spack view --dependencies yes -i local --exclude edepsim  \
      add edepsim@master+opticks cxxstd=17
git clone https://github.com/ClarkMcGrew/edep-sim.git

./local/bin/cmake -S edep-sim -B edep-sim-build \
                  -DCMAKE_INSTALL_PREFIX=$PWD/local \
                  -DCMAKE_INSTALL_RPATH='$ORIGIN/../lib' \
                  -DCMAKE_INSTALL_RPATH_USE_LINK_PATH=ON
./local/bin/cmake --build edep-sim-build
./local/bin/cmake --install edep-sim-build
./local/bin/edep-sim
```

- Build stays outside of source area (inside is okay if you prefer).
- The "view" provides `local/{bin,lib,include}/` with symlinks to all dependencies.
- We also install our developer build into the view directory.
- The RPATH directives mean we don't have to mess with `LD_LIBRARY_PATH`.
- May want to `--exclude eicopticks` if you also want to develop eic-opticks.
- Can `rm -rf local/` and remake, at will.

Developer shell environment with Direnv

By default, a Spack View does not provide any **shell environment**.
My favorite is to make `hack/.envrc` file with this line:

```
load_prefix "$PWD/local"
```

Can include bash-like variable settings as well. Then add to your `~/.bashrc` or similar

```
eval $(direnv hook bash)
```

Then when you `cd` to your `hack/` directory the environment settings are applied automatically and removed when you `cd` elsewhere. Eg, can now run without giving explicit path:

```
$ cmake [....]
$ edep-sim [....]
```

Initial development plan

I think we do not yet know best way to integrate and suggest a two phase approach.

Phase 1: a quick, throw-away hack with no care for "design".

1. Fork `https://github.com/ClarkMcGrew/edep-sim.git`
2. Replace "vanilla" G4 optical with calls to `G4CXOpticks`.

In this process we will learn both edep-sim and eic-opticks / `G4CXOpticks` designs.

Phase 2: factored design with PR back to origin

1. Develop good design blessed by Clark and Gabor.
2. Add to edep-sim a compile-time dependency on eic-opticks, default: OFF.
3. Implement a run-time choice between "vanilla" G4 optical and use of `G4CXOpticks`.

Links

Repos

- https://github.com/ClarkMcGrew/edep-sim
- https://github.com/BNLNPPS/eic-opticks
- https://github.com/wirecell/edep-opticks-spack (spack repo)
- https://github.com/brettviren/winch (containers)
- https://github.com/BNLNPPS/esi-shell (containers)

DeepWiki repo summaries

- https://deepwiki.com/ClarkMcGrew/edep-sim
- https://deepwiki.com/BNLNPPS/eic-opticks

Opticks upstream

- https://simoncblyth.bitbucket.io