# Spack Planning - Bug #24271

## spack install cetlib@develop issue

04/03/2020 01:54 PM - Eric Flumerfelt

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **Start date:** | 04/03/2020 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | **Spent time:** | 0.00 hour |

### Description

I'm trying to use Spack as the baseline package manager for the new DUNE DAQ Application Framework. So far, we have decided on 3 dependencies: Boost, TRACE, and cetlib (for basic_plugin).

Here's my procedure:

```
git clone https://github.com/spack/spack.git
cd spack
echo "export SPACK_ROOT=$PWD;. '$SPACK_ROOT/share/spack/setup-env.sh'" >setup-env.sh
spack mirror add  --scope site fnal https://spack-cache-1.fnal.gov/binaries/
spack buildcache keys --install --force
spack buildcache keys --trust
cd var/spack/repos
git clone https://cdcvs.fnal.gov/projects/spack-planning-spack_art spack_art
cd ../../../etc/spack
cp defaults/repos.yaml .
echo '  - $spack/var/spack/repos/spack_art' >>repos.yaml
cd ../..
spack install trace@3.15.08
spack install cetlib@develop
```

I'm currently getting

```
[eflumerf@ironvirt7 spack]$ spack install cetlib@develop
==> Error: trying to set variant "single_header" in package "catch2", but the package has no such variant [happened during concretization of cetlib@develop ^catch2@2.3.0:~single_header ^ncurses ^pkgconfig ^readline ^sqlite@3.8.2: ^zlib]
```

I've edited var/spack/repos/spack_art/packages/cetlib/package.py to remove the single_header variant from catch2, and this appears to build OK. Is there a better fix that I should do? Should I commit this change to the repo (maybe on a for-dune-appfwk branch?)

### History

#### #1 - 04/03/2020 01:55 PM - Eric Flumerfelt

Realized that was an old version of my procedure:

```
# Install Spack
git clone https://github.com/spack/spack.git
cd spack
echo "export SPACK_ROOT=$PWD;. \$SPACK_ROOT/share/spack/setup-env.sh" >setup-env.sh
source setup-env.sh
spack bootstrap

# OPTIONAL: Install FNAL Spack buildcache
spack mirror add  --scope site fnal https://spack-cache-1.fnal.gov/binaries/
spack buildcache keys --install --force
spack buildcache keys --trust

# Install spack_art repository for FNAL-developed products (TRACE & cetlib)
cd var/spack/repos
git clone https://cdcvs.fnal.gov/projects/spack-planning-spack_art spack_art
cd ../../../etc/spack
```

```
cp defaults/repos.yaml .
echo '  - $spack/var/spack/repos/spack_art' >>repos.yaml
cd ../..

# Install dependencies
spack install gcc@8.2.0
spack load gcc@8.2.0
spack compiler find
spack install trace@3.15.08 %gcc@8.2.0
spack install cetlib@MVP1a %gcc@8.2.0
```

**#2 - 04/26/2020 05:29 PM - Christopher Green**

*- Status changed from New to Feedback*

Sorry for the delay replying to this.

The recipes for FNAL packages are currently only tested with respect to the FNAL fork of Spack, which has not been updated with respect to the upstream repository in quite some time (about 8 months).

Apparently six months ago the catch recipe was renamed, catch2, along with some other changes, one of which is the removal of the `single_header` variant.

Short answer: you can get going again by replacing "catch" with "catch2" and removing "~single_header" in any depends_on line that mentions it.

In general, one cannot right now rely on any recipe in the spack_art repo working with anything other than the FNAL fork of Spack. In practice, I think this is currently the only issue of which I am aware.

Please let us know if this workaround is sufficient for you at this time.

**#3 - 04/26/2020 05:34 PM - Eric Flumerfelt**

That workaround works for me, and furthermore I'd like to have it in the repo so that if and when we need cetlib using vanilla spack, we can use the spack_art repo for the Fermilab products. (There may be more development along these lines in the future as we try and get more of the DAQ software Spack-compatible.)

Eric Flumerfelt wrote:

> I've edited var/spack/repos/spack_art/packages/cetlib/package.py to remove the single_header variant from catch2, and this appears to build OK. Is there a better fix that I should do? Should I commit this change to the repo (maybe on a for-dune-appfwk branch?)

**#4 - 04/27/2020 07:04 AM - Christopher Green**

A perfectly fine idea. You are now a developer, and should have write access to the spack_art repository within 15 min.

**#5 - 04/27/2020 07:04 AM - Christopher Green**

*- Status changed from Feedback to Resolved*