NuWa Installations with `dybinst`

Brett Viren

Physics Department



Daya Bay NuWa Offline Software Tutorial

Overview

1. Who is this for?
2. What gets installed?
3. How do I use it?
4. Are there any options?
5. What if something goes wrong?
6. What do I get in the end?
7. Some Details.

Who is this for?

You should be interested in `dybinst` if you:

- are responsible for an "official" Daya Bay cluster.
- want to install NuWa software on your own machines.
- are just curious.
- you think `dybinst` is dumb and you can do better!

What does dybinst install?

CMT the basis of our build and runtime setup

external packages all the 3^{rd} party support software, currently:

AIDA, Boost, CLHEP, GCCXML Geant4, GSL, HepMC, HepPDT, MySQL client, OpenMotif, OpenScientist, Python, ROOT, Xerces-C.

LHC projects software taken from LHC effort (lcgcmt, gaudi, lhcb)

dybgaudi the project holding our Daya Bay specific offline software.

How do I use it?

1. Go home - don't try this on official clusters, they do it for you.
   https://wiki.bnl.gov/dayabay/index.php?title=Cluster_Account_Setup
2. Locate 3-4GB of free space (some reuse with subsequent releases)
3. Get the "dybinst" script

```
DYBSVN=http://dayabay.ihep.ac.cn/svn/dybsvn
svn export $DYBSVN/installation/trunk/dybinst/dybinst
```

4. Run it:

```
./dybinst <release_version> all
```

5. Wait 3-4 hours depending on your CPU, disk and network speeds.
6. Use it! Local sites may have special scripts, but it all boils down to:

```
source NuWa-RELEASE/setup.sh
CMTPROJECTPATH=/path/to/your/Project:$CMTPROJECTPATH
cd /path/to/your/Project/SomePackage/cmt
cmt config
source setup.sh
```

Are there any options?

The running can be broken down by steps by choosing a step name instead of `all`. These should be in order:

- **cmt** just download and build cmt
- **checkout** just checkout the NuWa release from SVN (dybgaudi and LHC projects)
- **external** download and build the 3^{rd} party software
- **projects** build the NuWa release (dybgaudi, gaudi, lhcb, etc) projects

idempotent - repeated applications of a function have the same effect as one.

`dybinst` tries to be idempotent so re-running a previously successful step does not cause it to be truly run again.

What if something goes wrong?

It won't!!

But, seriously,

1. Check what `dybinst` prints to the screen
2. Check the latest `dybinst-YYYMMDD-HHMMSS.log` file
3. Ask on `theta13-offline@lbl.gov`

Currently supported platforms

- CentOs 4.5, GCC 3.4.6 (VT)
- Debian 4.0 (etch), GCC 4.0 (BNL)
- Debian unstable (sid), GCC 4.2 (RPI)
- Mac OS X 10.4, GCC 4.0 (BNL) (avoid 10.5 for now)
- Sci. Linux 3.0, GCC 3.2 (IHEP,PDSF)

What do I get in the end?

`NuWa-RELEASE/`	Daya Bay specific software
`lcgcmt/`	project holding 3^{rd} party CMT "glue" packages and other setup
`gaudi/`	project holding the subset of Gaudi we use
`lhcb/`	project holding the subset of LHCb software we use
`dybgaudi`	project holding our NuWa offline software.
`external/`	the 3^{rd} party binaries
`installation/`	`dybinst` support files for the installing the release
`trunk/`	for the SVN trunk
`inst-NuWa-RELEASE/`	for a release

And some others....

Details: how it works.

Normally installers do not need to know this, but for experts, know that `dybinst` is composed of:

`dybinst` main script, release neutral, delegates to:

`dybinst/scripts/dybinst-common.sh` holds parameters of the release and useful shell functions.

`dybinst/scripts/dybinst-*` a suite of scripts in the `installation` project. There is one for each install step.

`dybinst/scripts/PACKAGE.sh` one script to handle each 3^{rd} party software package.

Details: Extending/modifying `dybinst`

More for experts: guidelines for adding your own 3^{rd} party install scripts:

- they should be idempotent
- install version according to the value in `lcgcmt/LCG_Configuration/cmt/requirements` (may need to be added if new package)
- populate the `external/` area according to the `lcgcmt/LCG_Interfaces/PACKAGE/cmt/requirements` glue package definitions, write such a glue package if needed
- consider looking at the `geant4.sh` script for a good example.
- platform-specific settings go in `lcgcmt/LCG_Settings/cmt/requirements`

Future Planned Features.

- Clean up logging (no logs for simple `dybinst help`).
- Fine-grained access to building `external/` and the projects.
- Parallelizing the build wherever possible.
- Support for alternative download locations: CN and US caches
- Investigate binary distribution (via PacMan?)
- Suggestions for this list?

Volunteers welcome!

Resources

- Wiki topics, https://wiki.bnl.gov/dayabay/index.php
  - "Offline Software Installation" main manual
  - "Offline Tutorial:dybinst" more tutorial material
  - "Offline Tutorial:projects and packages" also describes briefly our and others projects that we use