

JLAB Common Environment

A quest for the least evil

Purpose: having a common environment developed across the halls.

Why? Environment Variables are used during setup, compilation and run time by lot of software. Halls cooperation will minimize the risks associated with it.

Software Environment Variables

Can be evil:

ROOTSYS

Why do we need ROOTBIN = ROOTSYS/bin?

Or ROOTLIB = ROOTSYS/lib

Then there are variations... ROOT_BIN

ROOT_LIB

ROOT_INC... shees!

Geant4: Env. Variables to the pre-compiler and to the compiler as well. 24 of them.

Software Environment Variables

Can be evil:

ROOTSYS

Why do we need ROOTBIN = ROOTSYS/bin?

Or ROOTLIB = ROOTSYS/lib

Then there are variations... ROOT_BIN

ROOT_LIB

ROOT_INC... shees!

Geant4: Env. Variables to the pre-compiler and to the compiler as well. 24 of them.

We can't get rid of these but we simplify things for users.

Back to Basis

Ideally, we need only ONE environment variable for ALL libraries:

JLAB_SOFTWARE

- > geant4
- > mysql
- > clhep
- > jana
- > gemc

Back to Basis

JLAB_SOFTWARE

```
> geant4
    include
    bin
    lib
    [...]
> mysql
> clhep
> jana
> gemc
```

All right, let's use this model.

Software Versioning

JLAB_SOFTWARE

>> geant4

> 4.9.3.p02

> 4.9.4.p01

include

bin

lib

[...]

>> mysql

>> clhep

>> jana

> 0.6.2

include

bin

lib

[...]

Version Number Below Software Name

Machine Architecture

Use the `osrelease.pl` script (Dave Lawrence) to determine

- OS version
- CPU architecture
- Compiler Version

This returns a string, like:

`Darwin_macosx10.6-i386-gcc4.2.1`

`Linux_CentOS5.3-x86_64-gcc4.1.2`

Machine Architecture

Use the osrelease.pl script (Dave Lawrence):

Darwin_macosx10.6-i386-gcc4.2.1

Linux_CentOS5.3-x86_64-gcc4.1.2

Put it at the top of the path:

- So that everything below looks like the “ideal”
- Want to avoid having multiple subdirectories for each software
- Keep Sources Separated (don't mess with the .o)

The Plan

For the vast majority of users:

One Environment Variable:

JLAB_ROOT

(On the Cue Machines:

/site/12gev_phys)

Users:

```
setenv JLAB_ROOT /site/12gev_phys  
source $JLAB_ROOT /ce/jlab.csh
```

Will:

1. Run osrelease.pl to recognize machine
2. Set up all the evil environment (including LD and DYLD paths)
3. Set default versions, unless user specifies otherwise
4. Output Log on Screen

System Supported:

(supported = installed at JLAB, with RPM and DEB packages available)

- CLHEP
- Xerces
- Geant4 (with GDML)
- Qt4
- ROOT
- EVIO
- GEMC
- JANA
- SCONS Scripts

Log Output

```
jlab15% ssh ifarm1101
```

```
> Common Environment
```

```
> Running as ungaro on ifarm1101
```

```
> OSRelease: Linux_CentOS5.3-x86_64-gcc4.1.2
```

```
> JLAB_SOFTWARE set to: /site/12gev_phys/Linux_CentOS5.3-x86_64-gcc4.1.2
```

```
> CLHEP Version: 2.1.0.1
```

```
> Geant4 Version: 4.9.4.p02
```

```
> mysql is installed in: /site/12gev_phys/Linux_CentOS5.3-x86_64-gcc4.1.2/mysql
```

```
> QTDIR Version: 4.7.1
```

```
> ROOT Version: 5.28
```

```
> XERCES Version: 3.1.1
```

```
> EVIO is installed in: $JLAB_SOFTWARE/evio
```

```
> GEMC is installed in: $JLAB_SOFTWARE/gemc
```

```
? Attention: No SCRAT installation found.
```

```
> BANKS is installed in: $JLAB_SOFTWARE/banks
```

```
> Scons Scripts are in: $JLAB_ROOT/scons_scripts
```

```
> JANA Version: 0.6.2
```

User Customization

1. Can set up OSRELEASE to be different than osrelease.pl

Useful for cross-compiling (e.g. 32 bit on 64 bit machines)

User Customization

2. Can set up a software version env. variable to “freeze” it

```
setenv GEANT4_VERSION 4.9.4.p01
```

The scripts will check for a user variable version before setting the default version.

The version is a string. So one could test “geant4_mauree” version.

User Customization

3. Any existing library setup will not be overwritten
If you have ROOTSYS, everything ROOT related won't be overwritten

(still not so sure about that... Make it optional?)

Suggested Locations

JLAB_ROOT

On the Cue Machines:

`/site/12gev_phys`

On User's Linux:

`/usr/local/jlab_software`

On User's Mac OS X:

`/sw/jlab_software`

Packages distributions

All the above software is also distributed by:

- **RPMS** (Fedora 15 64, RHEL5, SL4 and CentOS 64)
- **Deb** (Ubuntu10)
- **Deb** (Fink, Mac OS X)

The environment is distributed as a package as well. The only thing that changes is JLAB_ROOT, everything downstream of that is the same.

Environment package: “ce”.

Packages distributions

All the above software is also distributed by:

- **RPMS** (Fedora 15 64, RHEL5, SL4 and CentOS 64)
- **Deb** (Ubuntu10)
- **Deb** (Fink, Mac OS X)

The packages are distributed with their SOURCES as well. So users can recompile their software using the packages.

Summary

Common Environment minimizes users and developers problems, while still allowing for lot of flexibility

Installed at JLAB, trivial to install on users machine

Would be nice to cooperate on this across the Halls

CON: not for BASH (yet)