

# Common Environment for Software Packages

M. Ungaro, Z. Zhao

March 28, 2014

## Abstract

We present a model to organize software packages into releases. Shell scripts define the environment variables needed to run and compile vs the packages. Users can switch between releases and can select different versions of individual packages.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Features</b>	<b>2</b>
<b>3</b>	<b>Quick Start</b>	<b>2</b>
3.1	production or development version (JLab) . . . . .	2
3.2	Custom JLAB.VERSION and/or location . . . . .	2
3.3	Custom version of a package . . . . .	3
<b>4</b>	<b>The directory structure</b>	<b>3</b>
4.1	The OSRELEASE variable . . . . .	4
4.2	Screen log . . . . .	4
<b>5</b>	<b>Installation</b>	<b>5</b>

## 1 Introduction

We present a set of tools to define a Common Environment (ce) at Jefferson Lab. This environment can be used to run and/or compile software against the packages. The packages supported are:

- CLHEP
- XERCESS
- QTDIR
- Geant4
- EVIO
- GEMC
- JANA
- ROOT
- Build
- Banks

## 2 Features

The packages are organized under a general software release number, called `JLAB_VERSION`. For example `JLAB_VERSION=1.0` sets

- CLHEP to version 2.1.3.1
- Geant4 to version 4.9.6.p02
- QTDIR to version 4.8.5
- XERCES to version 3.1.1
- ROOT to version 5.34.13
- GEMC to version 1.0
- JANA to version 0.7.1p3
- Build to version 1.0
- EVIO to version 4.0
- Banks to version 0.9

The main features are:

- Users can select `JLAB_VERSION`s.
- Users can select packages versions independently.
- Users can select custom OS, CPU architecture and compiler version
- The environment can be used to run and compile vs all the packages
- Users can install the packages using the same structure on their personal laptop/desktop.
- The packages are tested and maintained on Mac and Linux flavors.

## 3 Quick Start

### 3.1 production or development version (JLab)

Users can select either version with:

- `source /site/12gev_phys/production.csh`
- `source /site/12gev_phys/devel.csh`

### 3.2 Custom `JLAB_VERSION` and/or location

Users can select the root directory containing all packages, and the `JLAB_VERSION`:

1. point `JLAB_ROOT` to the location of the software packages
2. choose a `JLAB_VERSION`
3. source the script that sets the environment

with these commands (a JLAB specific example is shown):

```
setenv JLAB_ROOT /site/12gev_phys
setenv JLAB_VERSION 1.0
source $JLAB_ROOT/$JLAB_VERSION/ce/jlab.csh
```

Available `JLAB_VERSION`s can be seen at [latest `JLAB\_VERSION`](#).

### 3.3 Custom version of a package

Users can select a custom version of a package by setting the environment variable `NAME_VERSION` before sourcing `jlab.csh`, where `NAME` is the package name capitalized (`ROOT`, `CLHEP`, `QT`, etc). For example to choose `geant4` version `9.6.p03` instead of the default `9.6.p02` users can do this:

```
setenv JLAB_ROOT /site/12gev_phys
setenv JLAB_VERSION devel
setenv GEANT4_VERSION 4.9.6.p03
source $JLAB_ROOT/$JLAB_VERSION/ce/jlab.csh
```

Any user defined `NAME_VERSION` will overwrite the default.

## 4 The directory structure

The software is placed into versions in `$JLAB_ROOT`. At JLab, `$JLAB_ROOT` is `/site/12gev_phys`. Inside each version there are:

- the “ce” directory contains the environment
- the “install” directory contains the installation scripts
- the “scons\_bm” directory contains the scons script to compile software against the packages
- the `OSRELEASE` contains the actual packages packages

An illustration of the organization is shown in Fig. 1.

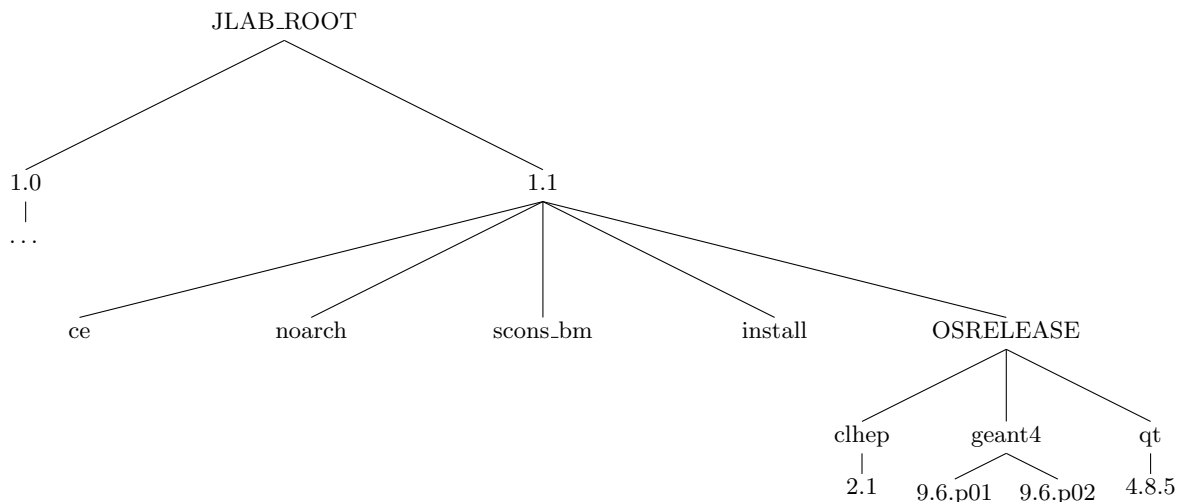


Fig. 1: The ce directory structure.

## 4.1 The OSRELEASE variable

Multiple OS, CPU architecture and compiler version are maintained using the OSRELEASE variable. OSRELEASE is setup automatically (by `ce/osrelease.pl`) however users can overwrite the default by setting the OSRELEASE environment variable before sourcing `jlab.csh`. Typical examples of OSRELEASE are shown below:

- `Linux_CentOS6.2-x86_64-gcc4.4.6`
- `Linux_RHEL6-x86_64-gcc4.4.7`
- `Darwin_macosx10.8-x86_64-gcc4.2.1`
- `Linux_Fedora20-x86_64-gcc4.8.2`
- `Linux_Ubuntu12.04-x86_64-gcc4.6`

## 4.2 Screen log

The scripts output on screen contain informations about the JLAB\_ROOT location, the JLAB\_VERSION, the OSRELEASE and the packages version. Below is a typical output.

```
> Common Environment Version: <production> (Tue, 25 Feb 2014)
> Running as ungaro on ifarm1101
> OS Release: Linux_CentOS6.2-x86_64-gcc4.4.6
> JLAB_ROOT set to: /site/12gev_phys
> JLAB_SOFTWARE set to: /site/12gev_phys/production/Linux_CentOS6.2-x86_64-gcc4.4.6

> CLHEP version: 2.1.3.1
> XERCES version: 3.1.1
> QTDIR version: 4.8.5
> Geant4 version: 4.9.6.p02
> ROOT version: 5.34.13
> GEMC version: 1.8
> JANA version: 0.7.1p3
> Build version: 1.0
> EVIO version: 4.0
```

## 5 Installation

Users can install the supported packages on their laptop/desktop using the installation scripts inside “install”:

```
cd $JLAB_ROOT/$JLAB_VERSION/install
```

```
./go_clhep  
./go_xercesc  
./go_mysql  
./go_qt4  
./go_geant4  
./go_sconsscript  
./go_evio  
./go_gemc
```

Optional: JLab magnetic field, cern ROOT, JLab banks library, JANA

```
./go_fields  
./go_root  
./go_banks  
./go_jana
```

A complete set of instructions and requirement for installation can be found [here](#).