

ALS-U EPICS Environment

Jeong Han Lee
jeonglee@lbl.gov

This document only covers the Rocky Linux distribution, but it may work on Redhat variants as well. If you want to install it on Debian, you can install the packages correctly (i.e. package names are slightly different), then all other Makefile procedures remain the same.

Packages

The following is the repository configuration I did before installing any packages and asking the unix team.

```
Unset
dnf -y install dnf-plugins-core
dnf -y config-manager --set-enabled powertools
dnf -y install "epel-release"
```

The following packages are not mandatory for ALS, as they are already installed by default. However, if you are using your own Rocky Linux, please install them first before installing others. In the last chapter we will show you more comfortable packages for the development environment.

```
Unset
dnf -y install git sudo
```

Here are some packages that are required for the ALS-U EPICS environment

```
Unset
dnf -y install \\\
tree which autoconf libtool automake re2c graphviz flex-devel patch
readline-devel libXt-devel libXp-devel libXmu-devel libXpm-devel motif-devel
gcc-c++ ncurses-devel perl-devel net-snmp net-snmp-utils net-snmp-devel
libzip-devel libusb-devel platform-python-devel boost-devel pcre-devel
libcurl-devel libxml2-devel hdf5-devel netcdf-devel libtiff-devel
libjpeg-turbo-devel libevent-devel libpng-devel libusbx-devel systemd-devel
```

```
libtirpc-devel libtirpc rpcgen re2c libusb-devel libusb python3-devel cmake  
libssh2-devel libssh2
```

I cannot install some packages because of the proxy configuration. I have asked the Unix team to set it up. For example,

```
Unset  
net-snmp net-snmp-utils net-snmp-devel boost-devel hdf5-devel
```

ULDAQ Requirements for EPICS measComp module

For the ALS-U Environment, we reserve the vendor specific library within the `INSTALL_LOCATION/vendor` Under `vendor` path, `include` and `lib` will be used for all not-system libraries and headers. We use the `INSTALL_LOCATION` as `/usr/local/epics/alsu`.

```
Unset  
$ git clone https://github.com/jeonghanlee/uldaq-env  
$ cd uldaq-env  
uldaq-env (master)$ echo "INSTALL_LOCATION=/usr/local/epics/alsu/vendor" >  
configure/CONFIG_SITE.local  
uldaq-env (master)$ make init  
uldaq-env (master)$ make conf  
uldaq-env (master)$ make build  
uldaq-env (master)$ make install  
uldaq-env (master)$ make exist  
tree -aL 1 /usr/local/epics/alsu/vendor  
/usr/local/epics/alsu/vendor  
├─ include  
└─ lib
```

Python packages

Rocky 8

```
Unset
$ sudo python3 -m pip install numpy nose2
```

Rocky 9

```
Unset
$ sudo python3 -m pip install numpy==1.19.5 nose2
```

EPICS Environment

There are three variables we should define.

- **EPICS_TS_NTP_INET** : I don't think we need to set this up if we are using any Linux host, but it doesn't hurt. If you are not on the ALS network, you can use `time.google.com` or any other NTP server near your location instead of `tic.lbl.gov`.
- **VENDOR_ULDAQ_PATH** : This is what you see in the above
- **INSTALL_LOCATION** : This is where the EPICS will be. You must have the write permission to this path. We use the **INSTALL_LOCATION** as `/usr/local/epics/alsu`. If you don't define it, it will use `${HOME}/epics` will be used as the default location. And only if your path does not contain `epics`, the `epics` path will also be added to your path as well.

We are still in Release Candidate mode, so please use the master branch with the latest commit.

```
Unset
$ git clone https://github.com/jeonghanlee/EPICS-env.git
$ cd EPICS-env
EPICS-env (master)$ echo "EPICS_TS_NTP_INET=tic.lbl.gov" >
configure/RELEASE.local
```

```
EPICS-env (master)$ echo "VENDOR_ULDAQ_PATH=/usr/local/epics/alsu/vendor" >>
configure/RELEASE.local
EPICS-env (master)$ echo "INSTALL_LOCATION=/usr/local/epics/alsu" >
configure/CONFIG_SITE.local
```

```
Unset
EPICS-env (master)$ make init
EPICS-env (master)$ make patch
EPICS-env (master)$ make conf
EPICS-env (master)$ make build
EPICS-env (master)$ make install
EPICS-env (master)$ make symlinks
```

```
Unset
$ source /usr/local/epics/alsu/1.1.0/rocky-8.8/7.0.7/setEpicsEnv.bash

$ softIoc
$ softIocPVA
$ softIocPVX
$ pvget -h
$ pvxget -h
$ caget -h
```

The SNMP mib files for Rocky 8/9

To easily troubleshoot any SNMP connection, I asked Kuldeep to check the following external files for the SNMP MIBs files. He approved the installation, so I installed it on appdev, which is the development server at ALS.

```
Unset
$ git clone https://github.com/jeonghanlee/snmp-mibs-downloader-env
$ cd snmp-mibs-downloader-env
$ snmp-mibs-downloader-env (master)$ make init
$ snmp-mibs-downloader-env (master)$ make install
$ snmp-mibs-downloader-env (master)$ make get
```

Additional Packages

Here is the list of additional packages we believe can help developers with their IOC development and deployment process.

Unset

```
dnf -y install vim bash-completion
```

Appendix A - Two Versions of A Module

This chapter shows how to add the different versions of the modules to the existing EPICS environment. For the production environment, the maintainer will only do this if there are specific requirements. This guide will help someone who wants to set up the environment locally for their test environment.

We select the `pvxs` module as an example. In the scenario, the existing environment, we have `pvxs 1.3.1`. However, it has a bug found later. So, we would like to upgrade the latest one `647775e`, AKA `1.3.2a`.

Edit RELEASE

Please open `configure/RELEASE`, and add the latest commit id in `SRC_TAG_PVXS` and `SRC_VER_PVXS` as follows:

```
Unset
SRC_TAG_PVXS:=647775e
SRC_VER_PVXS:=647775e
```

Follow the procedure

```
Unset
$ rm -rf pvxs-src          ## Remove the existing clone folder
$ make init.modules        ## Require init.modules NOT init.pvxs
$ make patch.pvxs.apply    ## <<<< This step is for a pvxs specific rule
$ make conf.pvxs
$ make build.pvxs
$ make install.pvxs
$ make symlinks
```

Check the versions

The environment has the following command like `make exist`. It helps users to see our installation folder. You can check if there are two PVXS in the environment, and the default is for the latest version that we just installed.

```
Unset
$ make exist
```

```
└─ pvxs -> ./pvxs-647775e
|   └─ pvxs-1.3.1
|   └─ pvxs-647775e
```