EUROPEAN SPALLATION SOURCE

e3 : ESS EPICS Environment

Jeong Han Lee

Integrated Control System Division
ESS, Sweden

https://www.europeanspallationsource.se
March 22, 2019

Presentations about e3

Since I am going to skip the important motivation why e3 was developed in this presentation, it is highly recommended to check the existent and previous presentations in CHESS.

Check the early presentations about e3

- 1st presentation at 2017-12-14 : ESS-0214598
- 2nd presentation at 2018-05-09 : ESS-0306067
- 3rd presentation at 2018-10-17 : ESS-0432106

Why Environment is Needed

EPICS Base

Version 7

ca
Com
dbCore
RecStd

Version 3

pvData
pvAccess
pvAccessCA
pvAccessIOC
qrsrv
pvDatabase
pvaClient
nt

Base

Why Environment is Needed

EPICS Modules

Community Modules  *Full Access*

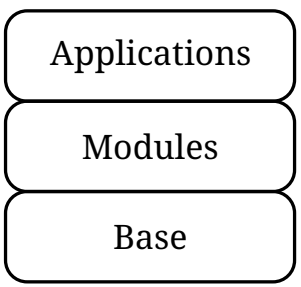
Site Specific Modules SLAC, Jefferson Lab, PSI, HZB Helmholtz Zentrum Berlin, SNS, FRIB, iter *Full & Limited Access*

Others Modules

Email, Private Communication, Sources, Code snippets

EPICS Applications

Loosely an application could be an EPICS IOC.



Why Environment is Needed

EUROPEAN SPALLATION SOURCE

Real E3 IOC for the Raster Scanning Magnet

Raster Scanning Magnet Specific application

Standalone EVR application

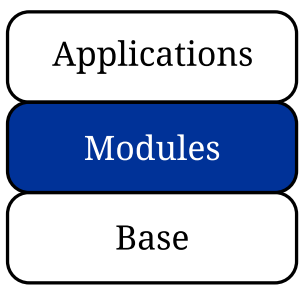
autosave 5.8.0 / community version (release version)

iocStats 18562f5 / community version (no release version)

mrfioc2 2.2.0-ess-rc1 / ESS customized version based on community 2.2.0

devlib2 2.9.0 / community version (release version)

Base 3.15.5 with several patch files



Why Environment is Needed

Complexity

- Each Site or Person follows the various ways to develop, maintain, and configure modules and applications.
- Each Site uses the different HW and SW architecture
- Site-wide subsystem be monitored by EPICS IOCs has its own requirements

Consistency

- Consistency for users and developers and even more for ESS Facility is the key to develop, operate, and maintain the control system from its initial conception through its retirement within the ESS's life cycle.
- ESS (or each site) needs its own Environment. (For FRIB, NSLS-II, Debian Packaging System / For ITER, CODAC / PSI, the original version of e3 / ...)
- **e3** was designed to achieve it.

Why E3 is needed : Global Prospective

EUROPEAN SPALLATION SOURCE

- Quality management of IOCs
  - EPICS full freedom : good for small groups
  - E3 limited freedom : good for ICS who has to provide the consistent environment to any stakeholders in ICS, Accelerator, Target, and Neutron Science
- Common source code management problems:
  - varying quality of modules (open source): code, documentation, & styles
  - version changes of base, modules, etc.
  - customized patch files, while synchronization with the EPICS community
  - platform variability
  - inconsistent version management overall in EPICS community
- Have to consider different EPICS users over ESS life time
  - advanced users: can manage their own IOC details
  - device integration focused (time limited) users: want to avoid low level development, compiling code etc.
  - less experienced users : benefit from pre-selection and prepared modules
  - core development users

Why E3 is needed : Local Prospective

EUROPEAN SPALLATION SOURCE

Users

- avoid re-building IOCs from scratch
- do not care about internal dependency among EPICS base, and modules)
- focus more on the IOC functionality and post-process of signals for each sub-system
- focus more on the user specific functionality (post-process, data analysis, user interface, and so on)
- transfer some IOC development effort shifts to a team of E3 Architects (currently, only one)
- use the ESS specific version rules consistently on EPICS base, and modules independent upon external sources
- avoid incompatible version combinations
- have the future migration process over EPICS base versions is less likely to cause problems

Desideratum for E3

European Spallation Source EPICS Environment Desideratum

Source Code Changes

- site, community, both patch files
- a time interval for possible merging activities
- rejection from community sources
- files, clone, fork, and branch ....

Maintain, maintain, and maintain

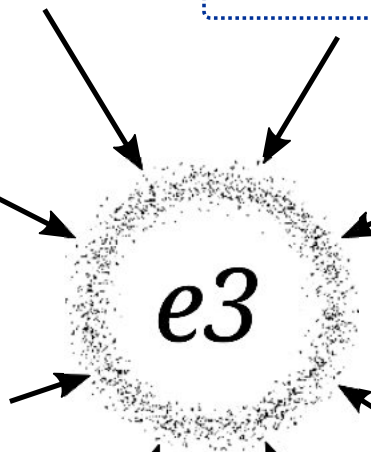
- clear structure to understand its dependency
- easily fire unused base, modules within the ESS life cycle
- duplicate the specific version of the e3 production in any places
- add new base, new modules into the production and into a development
- easily distinguish between e3 at t1 and e3 at t2

Release Version Numbers

- handle various version numbers (e.g., R1-0, v1.1, s7plc_1_4_0, no version)
- define ESS version for all of them (1.0.0, test, ae5d083, jhlee-testing)

Increase Degree of Freedom for Users

- Allow to have multiple e3 versions in a single host with EEE and the standard EPICS env.
- Allow to install e3 in any other Linux flavors (CentOS, Debian, Ubuntu, Fedora, Raspbian, ...)
- Allow to setup the standard EPICS env with e3
- Allow to have the entire setup locally



Disk and Network Resources

- Separate where src files from installed files in order to save disk and network resources (e3 source files ~ 2 GiB)

Quantized Integration / Deployment Continuously

- Deploy only the quantized version of the combination of all components (base, require, all other modules), since EPICS base is the long-standing one.
- Save resources to resolve potential overheads and to focus IOC functionality.

Users, Users, and Users

- want to run only working IOCs
- want to integrate an existent EPICS module into e3
- want to develop a module within e3
- want to develop a non-existent EPICS module with EPICS, then to integrate into e3
- want to develop a module based on the existent e3 modules

Mimicked EPICS Building System

- Mimic the EPICS building system
  * Makfile, configuration, rules, and so on
- In the future, will design the ESS rules, and configuration for the Standard EPICS building system.

EPICS IOC and E3 IOC

EPICS IOC	E3 IOC
Run makeBaseApp	
Define Base, Modules in RELEASE	
Add database and protocol files	Add database and protocol files
Update Makefile	
Build	
Edit st.cmd	Write st.cmd[1]
Run	Run

[1] define module

IOC : EPICS VS E3

EUROPEAN SPALLATION SOURCE

EPICS IOC

configure/RELEASE

```
EPICS_BASE=${EPICS_PATH}/epics-base/3.15.5
ASYN=${EPICS_MODULES}/asyn/4.33
STREAM=${EPICS_MODULES}/stream/2.7.7
devIocStats=${EPICS_MODULES}/iocStats/1856ef5
```

```
#!../../bin/linux-x86_64/gconpi

epicsEnvSet(P, "ICS")
epicsEnvSet(R, "E3TRNG")
epicsEnvSet("IOC",   "$(P):$(R)")
epicsEnvSet("IOCST", "$(IOC):IocStats")

epicsEnvSet("TOP","/home/jhlee/epics_env/epics-Apps/gconpi")
epicsEnvSet("STREAM_PROTOCOL_PATH", ".:${TOP}/db")

cd "${TOP}"

dbLoadDatabase "dbd/gconpi.dbd"
gconpi_registerRecordDeviceDriver pdbbase

drvAsynIPPortConfigure("CGONPI", "127.0.0.1:9999", 0, 0, 0)
dbLoadRecords("db/gconpi-stream.db", "SYSDEV=KAM:RAD1:,PORT=CGONPI")
dbLoadRecords("db/iocAdminSoft.db", "IOC=${IOCST}")

cd "${TOP}/iocBoot/${IOC}"
iocInit
```

E3 IOC

put the asyn dependency within stream dependency

```
require stream,2.7.7
require iocStats,1856ef5

epicsEnvSet(P, "ICS")
epicsEnvSet(R, "E3TRNG")
epicsEnvSet("IOC",   "$(P):$(R)")
epicsEnvSet("IOCST", "$(IOC):IocStats")
```

where the startup script exists

```
epicsEnvSet(TOP, "$(E3_CMD_TOP)")
epicsEnvSet("STREAM_PROTOCOL_PATH", ".:${TOP}/db")
```

put specific dbd load, and registerRecordDeviceDriver in behind scene

```
drvAsynIPPortConfigure("CGONPI", "127.0.0.1:9999", 0, 0, 0)
dbLoadRecords("${TOP}/db/gconpi-stream.db", "SYSDEV=KAM:RAD1:,PORT=CGONPI")
dbLoadRecords("iocAdminSoft.db","IOC=${IOCST}")
```

predefined module db can be searchable automatically

```
iocInit
```

E3 Anatomy and Requirements

EUROPEAN SPALLATION SOURCE

Building

source codes
configure
customize
patch files
compile
install

- proper version
- bi-sync with community's work
- track down changes
- kernel driver / configuration
- accessible headers
- system library
- source codes types
- how to get source codes
- supported OS
- special Makefile

Static

directory structure
environment
decommission
vendor library

- maintainable
- global environment
- local environment
- easy to duplicate
- demands from end-users
- define the dependency among base, modules, and others

Running

find
load
check
monitor

- where dependent system libraries
- where dependent modules
- where dbd and db files
- where IOC runs
- where the startup script
- iocsh.bash :
  collect necessary things
  to transfer to softIoc

E3 : Require from PSI, heavily customized one

- Require is the `EPICS` module
- ESS require[2] at `https://github.com/icshwi/require-ess`
- synced with latest changes of the PSI one
- to gain 10+ years experience of PSI, and customize it to meet the our own requirements

E3 Anatomy	**PSI**	**ESS**
Building & Static		E3 building system
Static & Running	require	require-ess†
Building	driver.Makefile	driver.Makefile†
Running	iocsh	iocsh.bash ‡

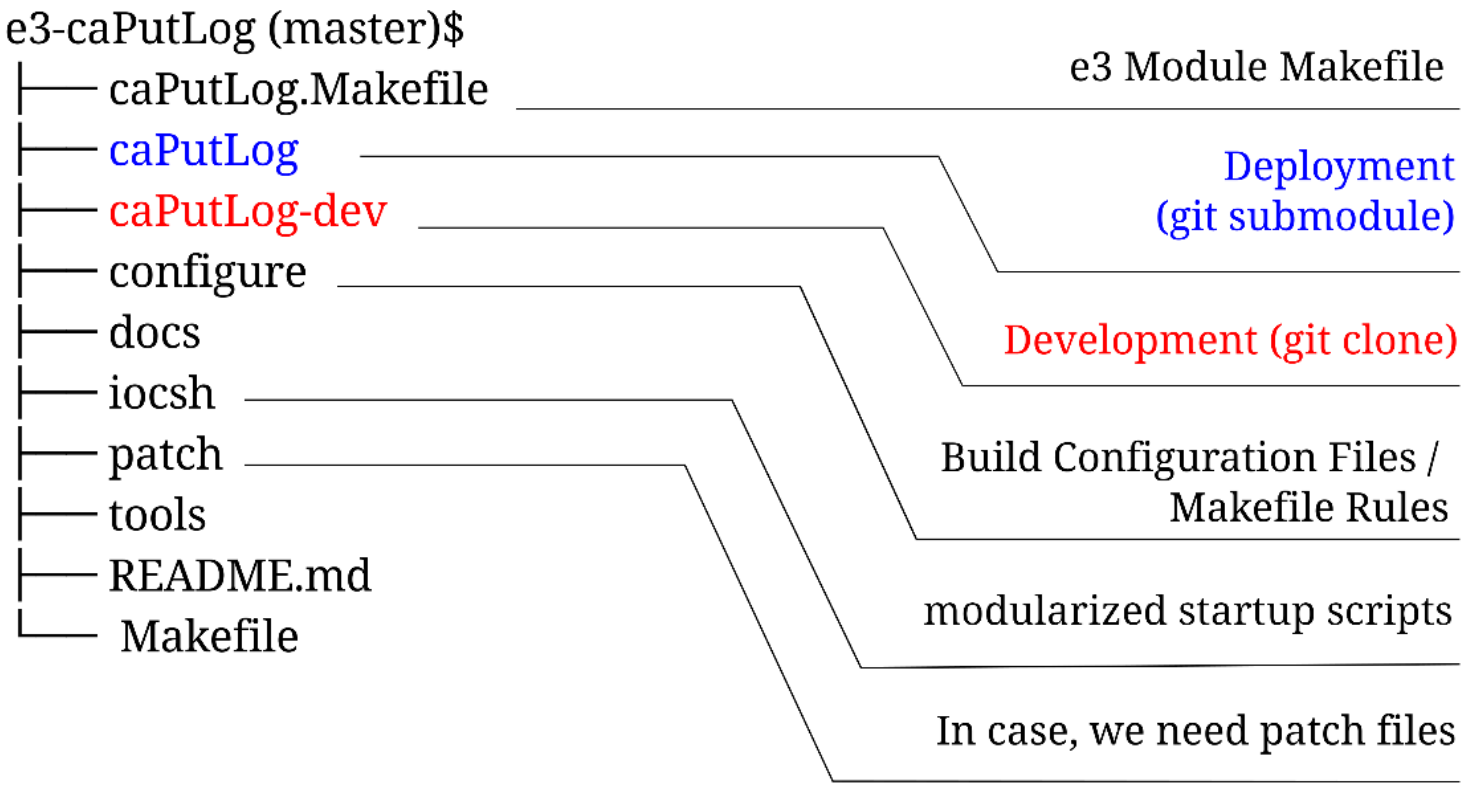
†: customized one
‡: rewritten completely

[2] from the PSI require `https://github.com/paulscherrerinstitute/require`

Short Introduction of E3 FE Template

EUROPEAN SPALLATION SOURCE

A Real Example e3 Structure for caPutLog



Short Introduction of E3 FE Template Generator

EUROPEAN SPALLATION SOURCE

Complicate?

```
$ bash e3TemplateGenerator.bash -m modules_conf/caputlog.conf
```

caputlog.conf

```
EPICS_MODULE_NAME:=caPutLog
EPICS_MODULE_URL:=https://github.com/epics-modules
E3_TARGET_URL:=https://github.com/icshwi
E3_MODULE_SRC_PATH:=caPutLog
```

e3 Tools

`https://github.com/icshwi/e3-tools`

- e3 Template Generator
- Linux RT PREEMPT Kernel configuration tool
- Others

e3 startup scripits

EUROPEAN SPALLATION SOURCE

Examples for

e3 Startup Scripts

```
require ess,0.0.1
require caPutLog,3.6.0
require iocStats,ae5d083
require stream,2.7.14p

epicsEnvSet("SEC", "SEC")
epicsEnvSet("SUB", "SUB01")
epicsEnvSet("P", "$(SEC)-$(SUB):")
epicsEnvSet("DIS", "DIS")
epicsEnvSet("DEV", "DEV-01")
epicsEnvSet("R", "$(DIS)-$(DEV)")
epicsEnvSet("IOCNAME", "$(P)$(R)")

epicsEnvSet("TOP",         "$(E3_CMD_TOP)/../..")
epicsEnvSet("ESS_TOP",     "$(TOP)/e3-ess")

epicsEnvSet("LOG_PORT", "5548")
epicsEnvSet("LOG_DEST", "10.4.4.30")

loadIocsh("accessSecurityGroup.iocsh", "ASG_PATH=$(ESS_TOP)/template, ASG_FILE=unrestricted_access.asg")
loadIocsh("iocStats.iocsh", "IOCNAME=$(P)")
loadIocsh("iocLog.iocsh", "IOCNAME=$(IOCNAME), LOG_INET=$(LOG_DEST), LOG_INET_PORT=$(LOG_PORT)")
loadIocsh("caPutLog.iocsh", "IOCNAME=$(IOCNAME), LOG_INET=$(LOG_DEST), LOG_INET_PORT=$(LOG_PORT)")

dbLoadRecords("$(E3_CMD_TOP)/sum.db","INST=$(IOCNAME)-md")

iocInit
```

```
require iocStats,ae5d083
require recsync,1.3.0
require autosave,5.9.0

epicsEnvSet("SEC", "SEC")
epicsEnvSet("SUB", "SUB01")
epicsEnvSet("P", "$(SEC)-$(SUB):")
epicsEnvSet("DIS", "DIS")
epicsEnvSet("DEV", "DEV-01")
epicsEnvSet("R", "$(DIS)-$(DEV)")
epicsEnvSet("IOCNAME", "$(P)$(R)")

iocshLoad("$(iocStats_DIR)/iocStats.iocsh", "IOCNAME=$(IOCNAME)")
iocshLoad("$(recsync_DIR)/recsync.iocsh",  "IOCNAME=$(IOCNAME)")
iocshLoad("$(autosave_DIR)/autosave.iocsh", "IOCNAME=$(R), AS_TOP=/tmp")

iocInit
```

Deployment Mode vs Development on E3

EUROPEAN SPALLATION SOURCE

Type	Deployment	Development
Configuration	RELEASE CONFIG_MODULE	RELEASE_DEV CONFIG_MODULE_DEV
Build Commands	make init make build make install make env make uninstall make rebuild	make devinit make devbuild make devinstall make devenv make devuninstall make devrebuild

Where E3 is

EUROPEAN SPALLATION SOURCE

Clone It Today!

- `git clone https://github.com/icshwi/e3`

Building and Running Tested on

- CentOS
- Debian
- Raspbian Stretch
- Ubuntu
- LinuxMint
- Fedora

E3 Status

Base Support List

- Base 3.15.5 / 3.15.6
- Base 7.0.1.1 / 7.0.2

Modules Support List

- Almost all EPICS modules (iocStats, autosave, caPutLog, asyn, busy, modbus, sequencer, sscan, std, ip, calc, delaygen, stream, s7plc, MCoreUtils, recsyn, devlib2, mrfioc2, keypress, sysfs, symbolname, memDisply, i2cDev, exprtk ....)
- Area Detector (ADSupport, ADCore, ADSimDetector, ADAndor, ADAndor3, ADPointGrey, ADProsilica, ADPluginEdge, ...)
- Ethercat Modules (ecmc, motor, ethercatmc, ecmctraining)
- Full IFC Specific and LLRF Specific Modules
- Some BI Modules

Outlook

Future ...

- gcc 8.2, Kernel 4.14 LTS
- Integrate with the CI integration for the production

Questions?

EUROPEAN SPALLATION SOURCE

`Hmm, control, control. You must learn control!`

Yoda (*The Empire Strikes Back*)

Serdecznie dziękuje!

Tack!

감사합니다!

Thank you!

☺