E3 (E trēs) : ESS EPICS Environment

Jeong Han Lee

Integrated Control System Division
ESS, Sweden

https://www.europeanspallationsource.se
June 14, 2018

Why Environment is Needed

EPICS Base

Version 7

core
ca
database
libCom

Version 3

normativeTypes
pvAccess
pvData
pvDatabase
pva2pva
pvaClient

Base

Why Environment is Needed

EPICS Modules

Community Modules  *Full Access*

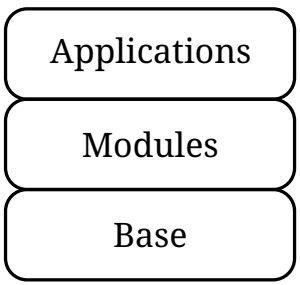
Site Specific Modules SLAC PSI SNS FRIB iter Jefferson Lab HZB Helmholtz Zentrum Berlin *Full & Limited Access*

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

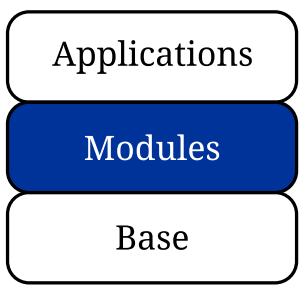
autosave 5.8.0 / community version (release version)

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

EUROPEAN SPALLATION SOURCE

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

- ESS (or each site) needs its own Environment. (Debian Packaging System for FRIB, NSLS-II, others / CODAC for Iter / E3 Origin for PSI / ...)
- Consistency for users and developers and even more for ESS Facility is the key to maintain the control system in the long term.
- **E3** was designed to achieve it in different user cases in terms of EPICS base, modules, applications, and others.

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

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

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

EUROPEAN SPALLATION SOURCE

- 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`

Warning ..

The following few pages are not 100% correct, but partly correct.
However with many assumptions and ignorance on different technical aspects, I would like to show the complicated situations which we will see in the near future.
And I would like to show how significantly we can reduce its complexity with the current E3 system.

Han

EUROPEAN SPALLATION SOURCE

Base (μ), Require (ν), N Module (ρ)

- Assumption 1 : all versions (μ, ν, ρ, i) are compatible with each others
- Assumption 2 : $N = 3$, we have 3 different modules
- Assumption 3 : For 2 years, we have 2 base version ($\mu = 2$), 4 requires ($\nu = 4$), and 6 module version ($\rho, \sigma, \delta = 6$) per each modules
- $\mathrm{IOC}_{\mu\nu\rho\sigma\delta} = \mathrm{B}_\mu \times \mathrm{R}_\nu \times \mathrm{M}_\rho^1 \times \mathrm{M}_\sigma^2 \times \mathrm{M}_\delta^3$
- The total number of the likely existent IOC in our EPICS environment:

$$\sum_{\mu=1}^{2}\sum_{\nu=1}^{4}\sum_{\rho=1}^{6}\sum_{\sigma=1}^{6}\sum_{\delta=1}^{6}\mathrm{IOC}_{\mu\nu\rho\sigma\delta} = \mathbf{1782}$$

EUROPEAN SPALLATION SOURCE

Base (μ), Require (ν), N Module (ρ)

- Assumption 1 : all versions (μ, ν, ρ, i) are compatible with each others
- Assumption 2 : $N = 3$, we have 3 different modules
- $\mathrm{IOC}_{\mu\nu\rho\sigma\delta} = \mathrm{B}_\mu \times \mathrm{R}_\nu \times \mathrm{M}_\rho^1 \times \mathrm{M}_\sigma^2 \times \mathrm{M}_\delta^3$
- The total number of the likely existent IOC in our EPICS environment:

$$\sum_{\mu=1}^{2}\sum_{\nu=1}^{4}\sum_{\rho=1}^{6}\sum_{\sigma=1}^{6}\sum_{\delta=1}^{6}\mathrm{IOC}_{\mu\nu\rho\sigma\delta} = \mathbf{1782}$$

- Are we sure how to handle them in terms of disk space, network traffic, and so on?
- How can we drop old base, require, and module versions, which no one uses?
- The assumption 1 likely is not true, we have a lot of incompatible issues among all of them. How do we fix this?

EUROPEAN SPALLATION SOURCE

Base (μ), Require (ν), N Module (ρ)

- Assumption 1 : all versions (μ, ν, ρ, i) are compatible with each others
- Assumption 2 : $N = 3$, we have 3 different modules
- Assumption 3 : For 2 years, we have 2 base version ($\mu = 2$), 4 requires ($\nu = 4$), and 6 module version ($\rho, \sigma, \delta = 6$) per each modules
- $\mathrm{IOC}_{11\rho\sigma\delta} = \mathrm{B}_1 \times \mathrm{R}_1 \times \mathrm{M}_\rho^1 \times \mathrm{M}_\sigma^2 \times \mathrm{M}_\delta^3$
- The total number of the likely existent IOC in our EPICS environment:

$$\sum_{\rho=1}^{6} \sum_{\sigma=1}^{6} \sum_{\delta=1}^{6} \mathrm{IOC}_{11\rho\sigma\delta} = \mathbf{216}$$

EUROPEAN SPALLATION SOURCE

Base (μ), Require (ν), N Module (ρ)

- Assumption 1 : all versions (μ, ν, ρ, i) are compatible with each others
- Assumption 2 : $N = 3$, we have 3 different modules
- Assumption 3 : For 2 years, we have 2 base version ($\mu = 2$), 4 requires ($\nu = 4$), and 6 module version ($\rho, \sigma, \delta = 6$) per each modules
- $\mathrm{IOC}_{11\rho\sigma\delta} = \mathrm{B}_1 \times \mathrm{R}_1 \times \mathrm{M}_\rho^1 \times \mathrm{M}_\sigma^2 \times \mathrm{M}_\delta^3$
- The total number of the likely existent IOC in our EPICS environment:

$$\sum_{\rho=1}^{6} \sum_{\sigma=1}^{6} \sum_{\delta=1}^{6} \mathrm{IOC}_{11\rho\sigma\delta} = \mathbf{216}$$

- After decoupling the base and require from others, E3 structure has much less complexity than PSI one such as $\frac{216}{1782} \sim 0.12$. And we can drop old base and require easily.

E3 : Structure

EUROPEAN SPALLATION SOURCE

```
/epics/
├── [root    4.0K]  base-3.15.5/
└── [root    4.0K]  base-3.16.1/
```

E3 : Structure

```
/epics/
├── [root    4.0K]  base-3.15.5/
└── [root    4.0K]  base-3.16.1/
```

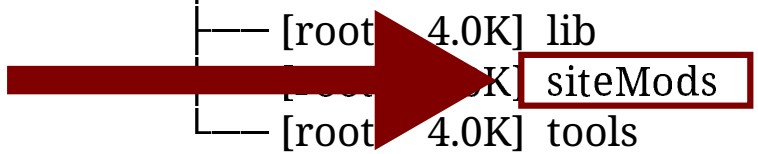
```
base-3.15.5/
├── [root    4.0K]  bin
├── [root    4.0K]  configure
├── [root    4.0K]  db
├── [root    4.0K]  dbd
├── [root    4.0K]  html
├── [root     12K]  include
├── [root    4.0K]  lib
├── [root    4.0K]  require
├── [root    4.0K]  startup
└── [root    4.0K]  templates
```

Require module has the dependency upon EPICS Base, so we put them under each EPICS Base version

E3 : Structure

Each module has dependency upon a require module version, so we put them under a require version

```
base-3.15.5/require/0.0.1/
├── [root    4.0K]  bin
├── [root    4.0K]  db
├── [root    4.0K]  dbd
├── [root    4.0K]  include
├── [root    4.0K]  lib
├── [root    4.0K]  siteMods
└── [root    4.0K]  tools
```



```
base-3.15.5/
├── [root    4.0K]  bin
├── [root    4.0K]  configure
├── [root    4.0K]  db
├── [root    4.0K]  dbd
├── [root    4.0K]  html
├── [root     12K]  include
├── [root    4.0K]  lib
├── [root    4.0K]  require
├── [root    4.0K]  startup
└── [root    4.0K]  templates
```

```
base-3.15.5/require/
├── [root    4.0K]  0.0.0
├── [root    4.0K]  0.0.1
└── [root    4.0K]  2.5.4
```

E3 : Structure

```
base-3.15.5/require/0.0.1/
├── [root     4.0K]  bin
│   ├── [root      20]  ess-env.conf
│   ├── [root     2.3K]  iocsh.bash
│   ├── [root     6.6K]  iocsh_functions
│   └── [root     2.8K]  setE3Env.bash
├── [root     4.0K]  db
│   └── [root      661]  moduleversion.template
├── [root     4.0K]  dbd
│   └── [root      146]  require.dbd
├── [root     4.0K]  include
│   ├── [root      390]  asprintf.h
│   ├── [root      697]  require.h
│   └── [root      177]  strdup.h
├── [root     4.0K]  lib
│   ├── [root     4.0K]  linux-ppc64e6500
│   │   ├── [root      78K]  librequire.so
│   │   └── [root       68]  require.dep
│   └── [root     4.0K]  linux-x86_64
│       ├── [root     141K]  librequire.so
│       └── [root       68]  require.dep
├── [root     4.0K]  siteMods
│   └── [root     4.0K]  iocStats
│       ├── [root     4.0K]  0.0.1
│       └── [root     4.0K]  develop
└── [root     4.0K]  tools
    ├── [jhlee     38K]  driver.makefile
    ├── [root     3.2K]  expandDBD.tcl
    └── [root     7.9K]  getVersion.tcl
```

iocsh.bash
setE3Env.bash
per a version of require

makefile, and other scripts
per a version of require

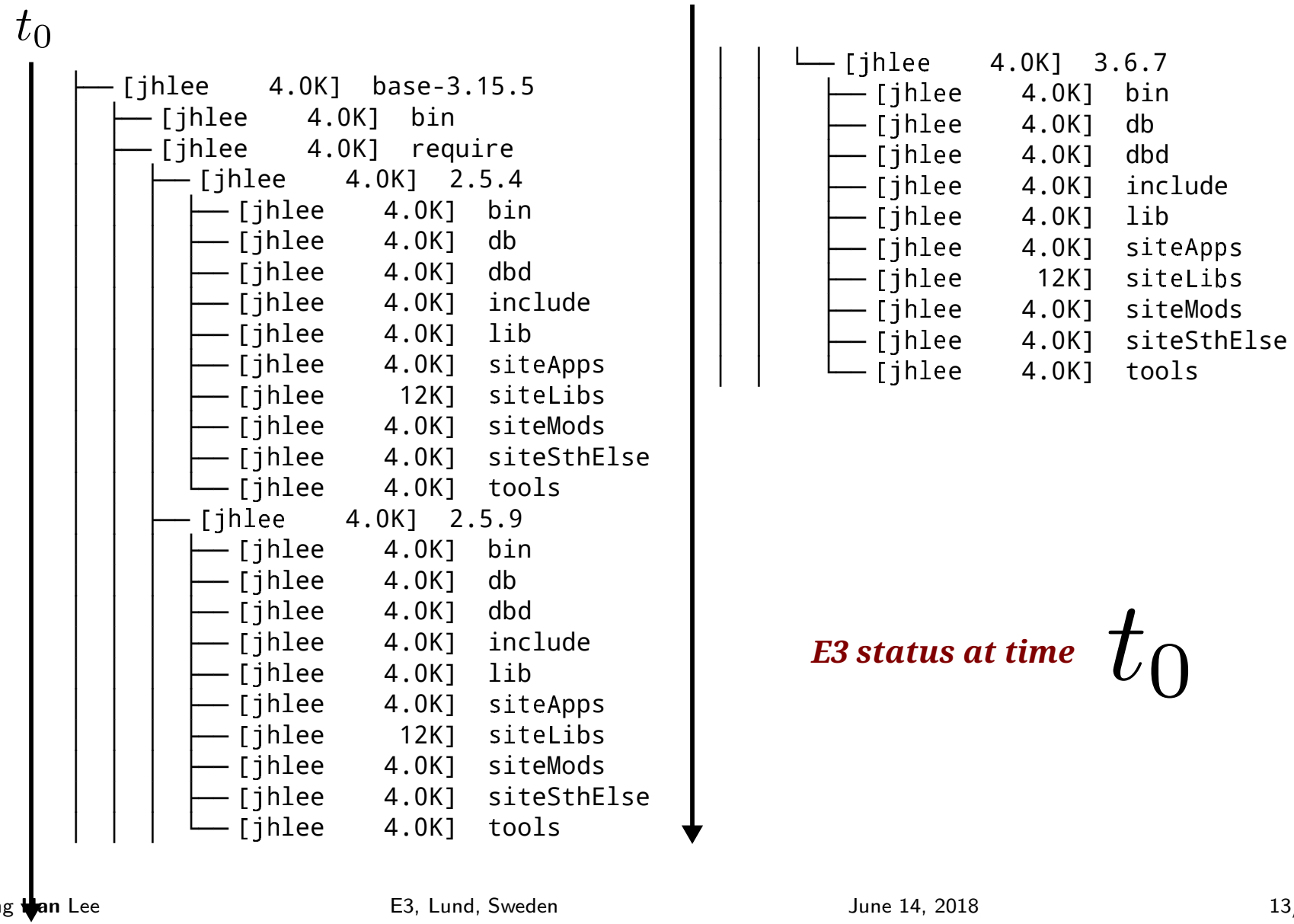
E3 : Structure

```
base-3.15.5/require/0.0.1/siteMods/
└── [root     4.0K]  iocStats
    ├── [root     4.0K]  0.0.1
    │   ├── [root     4.0K]  db
    │   ├── [root     4.0K]  dbd
    │   ├── [root     4.0K]  include
    │   └── [root     4.0K]  lib
    └── [root     4.0K]  develop
        ├── [root     4.0K]  db
        ├── [root     4.0K]  dbd
        ├── [root     4.0K]  include
        └── [root     4.0K]  lib
```

DB files belong to its version

```
base-3.15.5/require/0.0.1/siteMods/iocStats/0.0.1/
├── [root     4.0K]  db
│   ├── [root      2.0K]  access.db
│   ├── [root      1.1K]  iocCluster.template
│   ├── [root       157]  iocEnvVar.template
│   ├── [root      1.1K]  iocGeneralTime.template
│   ├── [root       632]  iocRTEMSOnly.template
│   ├── [root      2.5K]  iocRTOS.template
│   ├── [root       555]  iocScanMonSum.template
│   ├── [root       639]  iocScanMon.template
│   ├── [root      8.5K]  ioc.template
│   └── [root       611]  iocVxWorksOnly.template
├── [root     4.0K]  dbd
│   └── [root       418]  iocStats.dbd
├── [root     4.0K]  include
│   ├── [root      3.4K]  devIocStats.h
│   └── [root       946]  devIocStatsOSD.h
└── [root     4.0K]  lib
    ├── [root     4.0K]  linux-ppc64e6500
    │   ├── [root       31]  iocStats.dep
    │   └── [root      69K]  libiocStats.so
    └── [root     4.0K]  linux-x86_64
        ├── [root       31]  iocStats.dep
        └── [root     151K]  libiocStats.so
```

E3 : From t_0 to t_1



E3 status at time t_0

E3 : From t_0 to t_1

EUROPEAN SPALLATION SOURCE

t_0

```
├── [jhlee    4.0K]  base-3.15.5
│   ├── [jhlee    4.0K]  bin
│   ├── [jhlee    4.0K]  require
```

E3 Status at time t_1

We can drop old require versions in base-3.15.5, that means all old modules below them also we can drop.

Or, we can seperate old E3 easily, in case we can use only them to run old IOC, if the ioc has no issue with limited network and disk resources.

```
│   │   └── [jhlee    4.0K]  3.6.7
│   │       ├── [jhlee    4.0K]  bin
│   │       ├── [jhlee    4.0K]  db
│   │       ├── [jhlee    4.0K]  dbd
│   │       ├── [jhlee    4.0K]  include
│   │       ├── [jhlee    4.0K]  lib
│   │       ├── [jhlee    4.0K]  siteApps
│   │       ├── [jhlee     12K]  siteLibs
│   │       ├── [jhlee    4.0K]  siteMods
│   │       ├── [jhlee    4.0K]  siteSthElse
│   │       └── [jhlee    4.0K]  tools
├── [jhlee    4.0K]  base-3.16.1
│   ├── [jhlee    4.0K]  bin
│   ├── [jhlee    4.0K]  require
│   │   ├── [jhlee    4.0K]  2.5.9
│   │   ├── [jhlee    4.0K]  3.6.7
│   │   └── [jhlee    4.0K]  3.6.8
├── [jhlee    4.0K]  base-7.0.0
│   ├── [jhlee    4.0K]  bin
│   ├── [jhlee    4.0K]  require
│   │   ├── [jhlee    4.0K]  3.6.7
│   │   ├── [jhlee    4.0K]  4.0.0
│   │   └── [jhlee    4.0K]  5.0.0
```

t_1

E3 Stages

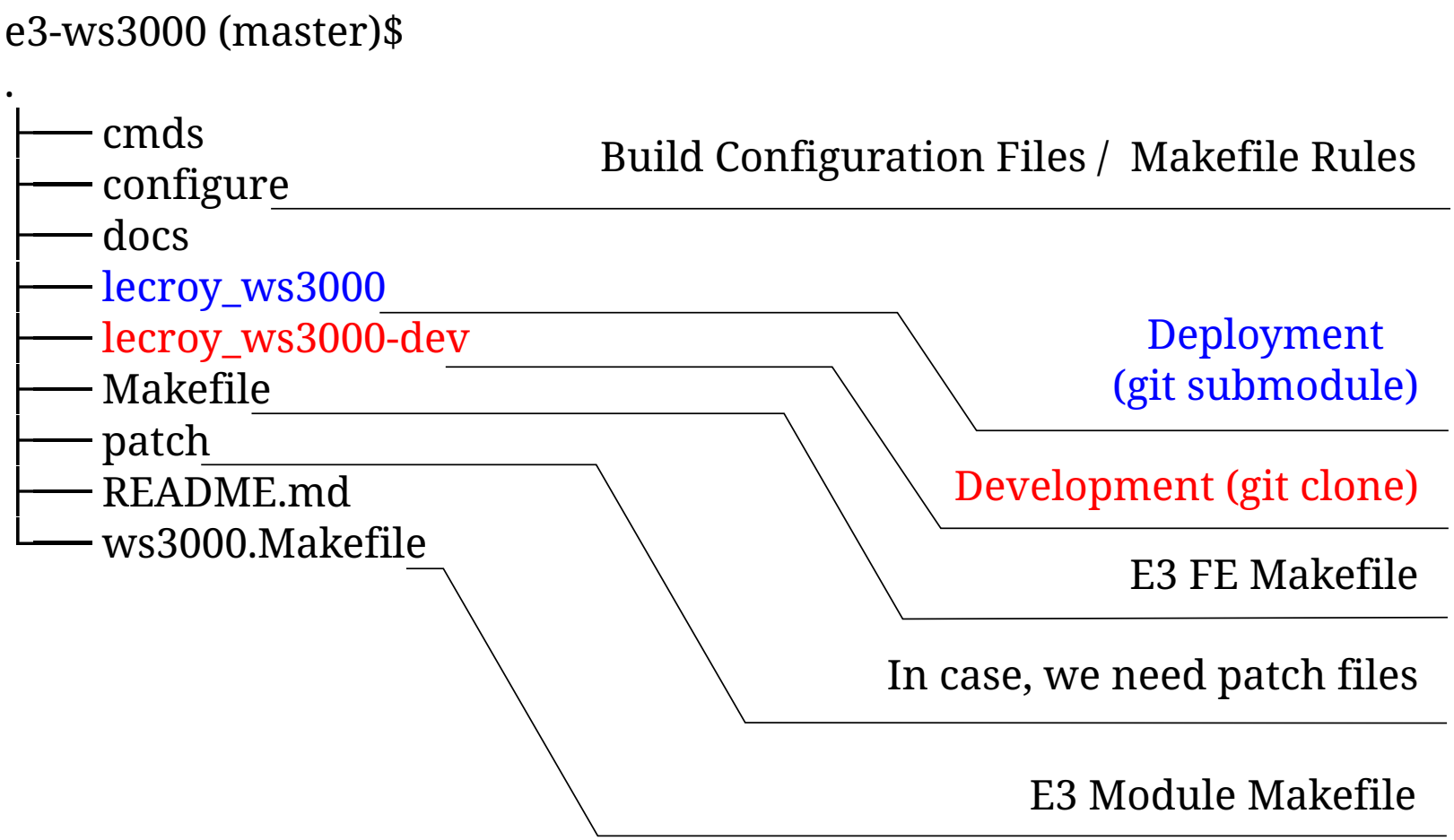
Current Stage : E3

- Building & Static : Redesigned the Building System (similar approach to EPICS Building System)
- Building & Running : Heavily modified PSI makefile, Replaced iocsh shell.
- Integration Test in progress (Timing, Motion, IFC platform, EtherCAT, Area Detector)
- Migration in progress (Beam Instrumtation, Low Level RF)

Next Stage : E3^{+}

- We can use the EPICS building system instead of the modified PSI makefile (driver.Makefile)
- Completely replace the PSI require module.
- Collect possible technical resources, design, and discuss its requirements
- Target Due Date : mid, 2020

A Real Example E3 FE Structure for Lecroy WaveStation EPICS IOC



EUROPEAN SPALLATION SOURCE

Complicate?

```
$ ./create_e3_modules.bash -m modules_conf/adpico8.conf
```

adpico8.conf
EPICS_MODULE_NAME:=adpico8
EPICS_MODULE_URL:=https://github.com/hinxx
E3_TARGET_URL:=https://github.com/icshwi
E3_MODULE_SRC_PATH:=adpico8

```
require ws3000,0.0.1
require iocStats,1856ef5
require autosave,5.9.0
require mrfioc2,2.2.0-rc1

epicsEnvSet("IOC","SUPERCYCLE")
epicsEnvSet("TOP",."")
epicsEnvSet(P, "usbtmc")
epicsEnvSet(R, "icslab")
epicsEnvSet(USBTMCPORT, "usbtmc0")
epicsEnvSet(WS3122PORT, "WS3122" )
epicsEnvSet(vendorNum,  "05ff")
epicsEnvSet(productNum, "0a21")
epicsEnvSet("EPICS_CA_MAX_ARRAY_BYTES","10000000")
epicsEnvSet("DEV1", "WSEVR")
epicsEnvSet("MainEvtCODE" "14")
epicsEnvSet("HeartBeatEvtCODE"   "122")
epicsEnvSet("ESSEvtClockRate"  "88.0525")

usbtmcConfigure("$(USBTMCPORT)", "0x$(vendorNum)", "0x$(productNum)")
drvWS3122Configure("$(WS3122PORT)", "$(USBTMCPORT)")
mrmEvrSetupPCI("$(DEV1)",  "01:00.0")

dbLoadRecords("asynRecord.db", "P=$(P), R=$(R),  PORT=$(USBTMCPORT), ADDR=0, OMAX=100,IMAX=100")
dbLoadRecords("WS3122Base.db", "P=$(P):,R=$(R):, PORT=$(WS3122PORT)")
dbLoadRecords("BasicWave.db",  "P=$(P):,R=$(R):, PORT=$(WS3122PORT)")
dbLoadRecords("BurstWave.db",  "P=$(P):,R=$(R):, PORT=$(WS3122PORT)")
dbLoadRecords("WS3122Cmds.db", "P=$(P):,R=$(R):, PORT=$(WS3122PORT)")
dbLoadRecords("iocAdminSoft.db", "IOC=$(P):$(R):IocStats")
dbLoadRecords("evr-pcie-300dc-ess.db","EVR=$(DEV1), SYS=$(IOC), D=$(DEV1), FEVT=$(ESSEvtClockRate)")

var evrMrmTimeNSOverflowThreshold 100000

< ${TOP}/save_restore_before_init.cmd
iocInit
< ${TOP}/save_restore_after_init.cmd

dbl > "${TOP}/${IOC}_PVs.list"

dbpf $(IOC)-$(DEV1):DlyGen0-Evt-Trig0-SP $(MainEvtCODE)
dbpf $(IOC)-$(DEV1):DlyGen0-Width-SP 1000 # time in us
dbpf $(IOC)-$(DEV1):OutFPUV02-Src-SP 0 # trigger from delay generator 0
dbpf $(IOC)-$(DEV1):OutFPUV03-Src-SP 0 # trigger from delay generator 0
```

$ iocsh.bash e3_ws3000_evr.cmd

The system is running
at ICS Tuna Lab
for the SuperCycle Test

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

What's New on E3

EUROPEAN SPALLATION SOURCE

- Rewrite iocsh.bash completely
- Dynamic env. setting per a Require version
- More than 50 modules were integrated into E3 (Timing System, IFC platforms, Motion, Area Detector, EPICS Common modules)

NO **MORE** Largest Module Version will be used when Building and Running. For example, in the case that MODULE A (**MA**) with the version `MA2.0` needs MODULE B (**MB**).

Building : Header Files

case 1 If the VERSION of **MB** with `MB1.0` is defined, it will be used to compile **MA**.

case 2 if not, it will use the latest (e.g., `1.2.3`) version to compile **MA**.

Running : Library Files

case 1 **MB** with `MB1.0` will be loaded when `require` **MA**,`MA2.0`

case 2 **MB** with `1.2.3` will be loaded when `require` **MA**,`MA2.0`

Failure Scenario : Still, but Less and Less

- Very small probability p_μ[3]on IOC CRASH of a module (so called Han) while loading if Han has more than one dependent module and the dependent modules have also dependencies.
- Failure Rate $= \prod_{\mu=1}^{n} p_\mu$, where n is the number of dependent modules, which have its own dependencies.
- Main reason is that two independent methods to select *A VERSION* exist in **Building** and RUNNING
- However, we can minimize that scenario with the restrict Release and Change plans.

[3] Actual number cannot be estimated because it comes from human behavior

Where E3 is

EUROPEAN SPALLATION SOURCE

Clone It Today!

- `git clone https://github.com/icshwi/e3`
- `https://github.com/icshwi/e3training` (under development)

Building and Running Tested on

- CentOS 7.4/7.5
- Debian 8/9

Building Tested on (not recommanded)

- Raspbian Stretch
- Ubuntu 14.04.05 LTS / 16.04.3 LTS / 17.10
- LinuxMint 18.3 (sylvia)
- Fedora 27 (Workstation Edition)

Outlook

Near Future ...

- Develop the E3 hand-on training
- Prepare the user guide for E3
- More integration tests on MTCA platform (EtherCAT, Timing, Motion, IFC ADC cards)

Chess Doc Number : ESS-xxxxxx

Questions?

EUROPEAN SPALLATION SOURCE

Much to learn you still have ... my old padawan. This is just the beginning!

Yoda

Tack!

감사합니다!

Thank you!

Dankeschön!

☺