EPICS manager Tasks

August, 2001

N. Yamamoto
KEKB control group
KEK, JAPAN

for EPICS seminar in IHEP

Contents

- EPICS license
- How to get EPICS source code
- Software needed to build EPICS
- EPICS configuration
- Naming convention for Channel/Record
- Version control of software
- Static database

EPICS License

http://www.aps.anl.gov/epics/aboutLegal.php says,

"EPICSis not publicly available software.

It is currently available free to public and non-profit research institutions on signing a *license agreement*.

If you are interested in joining the EPICS collaboration, send initial enquiries to Dave Gurd (mailto:Dave Gurd <gurd@lanl.gov>). "

There is a discussion to put EPICS R3.14 into public domain. But No decision was made yet(as far as I know).

How to get EPICS source code

From WWW

- Latest EPICS Release can be obtained from WWW. However, you need an account and password to access.

From CVS server

- ANL/APS maintains CVS remote server for EPICS. EPICS collaboration member institute may access this CVS server.
- Development version is also available on CVS.

You MUST sign the license agreement before you get an account for these repository.

Software needed to build EPICS

EPICS source code

- Base : EPICS core software CA, process DB,...
- Extensions : Applications

Other tools

- C *and* C++ compilers (gcc 2.95.x will work). :
- perl (version 5 or later) : used in Make-system.
- GNU make (gmake) : GNU version of make. Make-system assumes gmake behavior.
- Tornado Development Environment (commercial product of Winds Rever Systems, Inc.)
- Motif/lestif (for some applications in extensions).
- CVS/RCS : Concurrent Version control System. Required to get EPICS source code from CVS server at ANL.
- ssh(secure shell)/open ssh : EPICS CVS server requires ssh.

optional tools

- SWIG : Simple Wrapper and Interface Generator. Used in ChannelArchiver/casi.
- Tcl/Tk , Python : Used in ChannelArchiver/casi

EPICS configuration(R3.13)

configuration file for EPICS build

- ☑ base/config/CONFIG_SITE
- ☐ base/config/CONFIG_SITE_HOST_ARCH.$(HOST_ARCH)
- ☐ base/config/CONFIG_SITE.$(BUILD_TYPE)
- ☐ base/config/CONFIG_SITE.$(BUILD_TYPE).$(HOST_ARCH)
- ☐ base/config/CONFIG_SITE.$(BUILD_TYPE).$(TARGET_ARCH)

Runtime Environment

- ☐ base/config/CONFIG_ENV : defines default value of environment variables
- ☑ base/config/CONFIG_SITE_ENV : defines default value of site specific environment variables
- ☑ startup/Site.cshrc
- ☑ startup/Site.profile

EPICS configuration(R3.14alpha2)

configuration file for EPICS build

- ☑ base/configure/CONFIG_SITE
- ☑ base/configure/os/CONFIG_SITE.*<host>*.*<os>*-*<arch>*
- ☐ base/configure/os/CONFIG.*<host>*.*<os>*-*<arch>*

  if your host or target system is not supported in the standard EPICS distribution.

Runtime Environment

- ☐ base/configure/CONFIG_ENV : defines default value of environment variables
- ☑ base/configure/CONFIG_SITE_ENV : defines default value of *site specific* environment variables
- ☑ base/startup/Site.cshrc
- ☑ base/startup/Site.profile

Ref: EPICS Application Developers Guide Chapter 4: EPICS Build Facility

base/config/CONFIG_SITE

Host Architecture for cross compiler

- HOST_ARCH=$(shell /proj/epics/R313/startup/HostArch)
- CROSS_COMPILER_HOST_ARCHS = hp700, etc
- CROSS_COMPILER_TARGET_ARCHS=pcore750 pcore603 frc40 frc64

path of libraries and tools

- GNU_DIR : Tornado tool directory
- TCL_LIBRARY =/proj/local/$(HOST_ARCH)/lib/tcl7.4/ , ...

Various Switches

- TORNADO = YES or NO
- SHARED_LIBRARIES=YES or NO
- STATIC_BUILD=YES or NO
- etc.

CONFIG_SITE.XXX

- These files can override the setting in CONFIG_SITE
  - base/config/CONFIG_SITE_HOST_ARCH.$(HOST_ARCH)
  - base/config/CONFIG_SITE.$(BUILD_TYPE)
  - base/config/CONFIG_SITE.$(BUILD_TYPE).$(HOST_ARCH)
  - base/config/CONFIG_SITE.$(BUILD_TYPE).$(T_A)
- HOST_ARCH : Host architecture e.g. solaris, hp700, alpha, Linux ...
- BUILD_TYPE : Ether "Host" or "Vx"
- T_A: Target architecture, e.g. mv167, mv177, frc40, ...

Adding new Arch_type in EPICS(R3.13)

Adding new type of VME computer(IOC).

- Create "base/config/CONFIG.Vx.<new_arch_type>" file. (You can copy one of these files and modify it)
- Define ARCH_CLASS and CMPLR_SUFFIX in this configuration file.
  - CMPLR_SUFFIX = ppc
  - ARCH_CLASS = ppc
- Create "base/config/CONFIG_SITE.Vx.<new_arch_type>" file for site speicific configuration.
- add <new_arch_type> to:

  CROSS_COMPILER_TARGET_ARCHS

  in base/config/CONFIG_SITE

Environment Variable at run-time

CONFIG_ENV

- EPICS_CA_ADDR_LIST="host1 host2..." : send CA search packet to the host on the list
- EPICS_CA_AUTO_ADDR_LIST=YES : Find all network interface on the host and use them for UDP broadcast
- EPICS_CA_CONN_TMO=30.0 : CA connection timeout in second (should be > 2*EPICS_CA_BEACON_PERIOD)
- EPICS_CA_REPEATER_PORT=5065 : CaRepeater port
- EPICS_CA_SERVER_PORT=5064 : Ca Server port
- EPICS_IOC_LOG_PORT=7004 : iocLogServer port

CONFIG_SITE_ENV

- EPICS_TS_MIN_WEST =-540 : Time difference between localtime and UTC.
- EPICS_TS_NTP_INET =xxx.yyy.zzz.ttt : NTP server ip address
- EPICS_IOC_LOG_INET =xxx.yyy.zzz.ttt : iocLogServer ip address
- EPICS_IOC_LOG_FILE_NAME=/tmp/iocLog.text
- EPICS_IOC_LOG_FILE_LIMIT=1000000

IOC: Booting Up IOC

Setup boot parameters on IOC

- VxWorks Kernel
- IP address
- Host IP

Prepare a startup script on a host WS.

- Edit st.cmd in iocBoot/ioc<name>

BOOTP/DHCP can be used to get some of boot parameters at boot time.

IOC: Bumpless restart

Choose a set of record to recover operation status

Save value of these records at appropriate time of operation

- Periodic snap shot
- just after(or before) some operation

Data recovery

- at IOC boot (bump less restart)
- by operator

Tools for bumpless restart

- Save/Restore routines on IOC (LANL)
- EPICS Bumpless Restart (LBL) : uses Save/Restore routines
- Saveset Archiver & Restore : Save set Archiver get data from CA.

IOC:Synchronization of the system

EPICS TimeStamp

- EPICS record has a timestamp which is updated at record processing
- Several Ways to synchronize time stamp on IOCs
  - Event Reciever HW (ANL) delivers Time Stamp signal
  - Soft
    - Master IOC delivers Time Stamp to slave
    - NTP server

Event System

- EPICS event in a process database
  - SCAN and EVNT fields are used to process a record when EPICS event is raised
  - post_event() raise an event.
  - Event Record can be also used to raise an event
- Event Reciever HW (ANL)
- KEKB Event Transmitter – Receiver

System:Naming convention

A name of channel should be unique in one control system.

- A name of channel should be less than *28* characters in length
- A name of channel should have no embedded non-alphanumeric characters other than ":", "_", "-", "<" and ">"

Naming conventions *should be established prior to an EPICS database.*

System:KEKB naming convention

Channel Name has a form

XXYZZZZ:rrrrrrrrrr:ssss:pppp

- XX : HW group name (BT, VA, MG, RF, BM, CO, LI, BE, ...)
- Y : Beam line/Accelerator (A:AR, L:LER, H:HER, a: AR BT, e: HER-BT, ...)
- ZZZZ: Device types(PS, BPM, ...)
- ssss : optional. Hierarchical name of Device or property.
- pppp : property name. (STOP, STRT, SW, CUR, ...)

System:Version control

Changes in an application can be a source of trouble in a control system.

Version control system allow you to track down the changes in the source code and you to find the problematic changes.

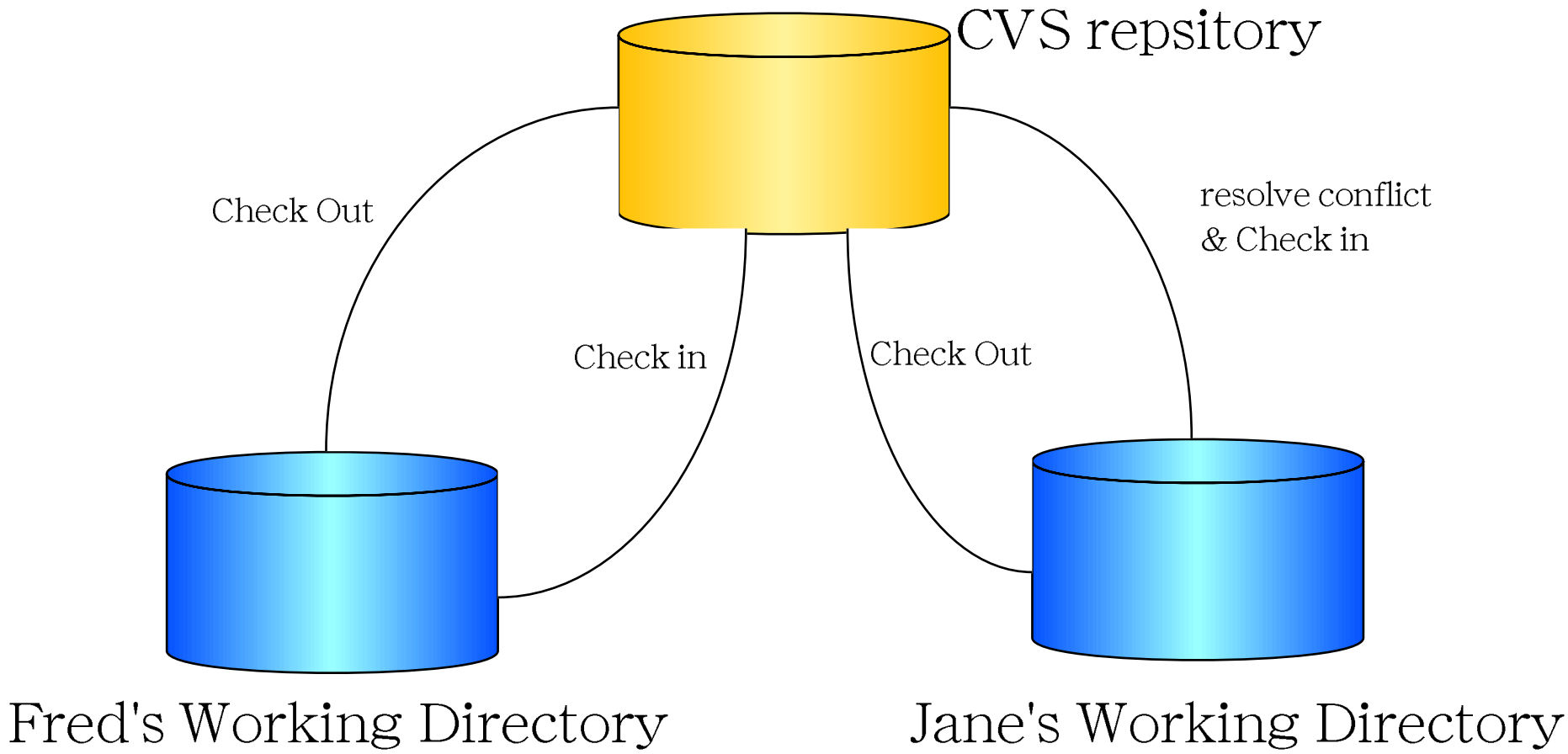
Version control using CVS is recommended

CVS:Concurrent Version control System

Check in/Check out

- CVS keeps current version of source code as well as a history of modification in the master file, CVS repository.
- An application developer checks out the version (s)he wants from repository to working directory. Later, (s)he will check in his(her) modification into repository.
- Multiple person can work on same repository at same time. There may be confilict between modification you made and modification by others. You need to resolve this confilit before "check in".

CVS:How it works



App:Setting up Application development Env.

makeBaseApp command setup directories for application development.

- App/
  - bin
    - <arch>
  - config
  - iocBoot
    - iocXXX
      - st.cmd : ioc boot up script
    - iocXXY
  - xxxApp
    - Db
    - src

makeBaseApp –t example XXX : creates XXXApp directory

makeBase App –i –t example XXX : create iocBoot/iocXXX directory

```
Usage:
./makeBaseApp.pl -l [options]
./makeBaseApp.pl -t type [options] app ...
             create application directories
./makeBaseApp.pl -i -t type [options] ioc ...
             create ioc boot directories
where
 app  Application name (the created directory will have
"App" appended to name)
 ioc  IOC name (the created directory will have "ioc"
prepended to name)

 -i       Specifies that ioc boot directories will be
generated
 -t type  Set the application type (-l for a list of valid
types)
          If not specified, type is taken from environment
          If not found in environment, "default" is used
 -T top   Set the template top directory (where the
application templates are)
          If not specified, top path is taken from
config/RELEASE
          If config does not exist, top path is taken from
environment
          If not found in environment, the templates from
EPICS base are used
 -l       List valid application types for this
installation
          If this is specified the other options are not
used
 -a arch  Set the IOC architecture (e.g. mv167)
          If not specified, you will be prompted
 -A arch  Set the IOC architecture (e.g. mv167)
          If not specified, you will be prompted
 -b base  Set the location of EPICS base (full path)
          If not specified, base path is taken from
config/RELEASE
          If config does not exist, base path is taken from
command
 -d       Verbose output (useful for debugging)

Environment:
EPICS_MBA_DEF_APP_TYPE  Application type you want to use as
default
EPICS_MBA_TEMPLATE_TOP  Template top directory

Example: Create exampleApp

<base>/bin/<arch>/makeBaseApp.pl -t example example
<base>/bin/<arch>/makeBaseApp.pl -i -t example example
```

makeBaseApp

You can create template files for your application tyep in

base/src/makeBaseApp/top/

base/src/makeBaseApp/top

- kekbApp
  - Db/
    - Makefile
    - Makefile.Host
    - db_APPNAME_1.db
    - db_APPNAME_2.substitutions
    - db_APPNAME_2.template
  - Makefile
  - src/
    - Makefile
    - Makefile.Host
    - Makefile.Vx
    - _APPNAME_Include.dbd
    - base.dbd
    - baseLIBOBJS
    - ca_APPNAME_.c
- kekbBoot
  - Makefile
  - ioc/
  - ioc/

App:Application Directory management.

- Application developers uses "makeBaseApp.pl" to setup his/her working directory.
- Rule should be set to integrate these application into the applications used in the operation.
- Use of CVS can be one of solution.
  - KEKB
    - HW_GROUP
      - XXXApp
      - iocBoot
      - bin

Data Storage

- A manager also need to prepare data storage shown right.
- It is alos important to organize directory structure of file system and to define naming convention of the files.

Applications

- CVS repository
- Official Application directory

Configurations

- Relational Database
- Configuration files
- Data snapshot

Data Archiving

- Archiving configuration :
- Logged Data : Can be HUGE. You must be prepared!

App : Tool selection

There are many ways to implement one functionality in EPICS.

- EPICS process database
  - with record links
  - with a new record type
- sequencer program with SNL
- Application on Host WS.
  - in C/C++
  - in scripting Languages

Tool should be selected taking accounts of:

- Time to develop application
- Performance/response time required
- Ease of long-time maintenance