SNS EPICS Application Development Environment and Control Software Product Control/Release Procedures

(Updated)
October 10, 2000

Johnny Tang, Steve Lewis, John Munro

References

1. "EPICS IOC Software Configuration Management" by Marty Kraimer, Andrew Johnson, Janet Anderson and Ralph Lange
   http://www.aps.anl.gov/asd/controls/epics/EpicsDocumentation/AppDevManuals/iocScm-3.13.2/index.html
2. "APS/ASD Application Source Release Control" by Marty Kraimer and Andrew Johnson
   http://www.aps.anl.gov/asd/controls/epics/EpicsDocumentation/EpicsGeneral/WorkInProgress/appRelease.html
3. "EPICS: GPIB Support for the Message Passing Facility" by Marty Kraimer
   http://www.aps.anl.gov/asd/people/mrk/epics/modules/bus/gpib/mpfGpib/index.html
4. "BESSY Software Distribution Area" by Ralph Lange
   http://www-csr.bessy.de/control/SoftDist/

Table of Contents

I. Introduction

This document describes the EPICS control system application development environment for the Spallation Neutron Source project (SNS/ADE) and a set of procedures for managing SNS control system IOC software products, EPICS extension products and non-EPICS control software products within the SNS/ADE. The design of SNS/ADE is based on the model described in "APS/ASD Application Source Release Control" and "EPICS IOC Software Configuration Management" that you should use as companion guides with this document.

In addition to the key features of the APS Source/Release Control system, the SNS/ADE uses one centralized CVS repository at ORNL and the developers at the other SNS collaborating labs use the remote CVS via secure shell to access the SNS central CVS repository. The SNS/ADE also manages EPICS extension software and other non-EPICS control software products.

SNS facility

The SNS facility is composed of five major subsections: front_end, linac, ring, target, and instrumentation. Each subsection is composed of several subsystems, such as RF, Magnet Power Supplies, Diagnostics, etc. Each subsystem is controlled by a separate set of IOCs. Each subsystem can be viewed as a major application that may be composed by several subapplications. One application also can be a support module that is needed in more then one major subsections. Such application will be viewed as a support application that will be shared by multiple applications.

Classes of SNS ADE users

SNS central EPICS administrator is responsible for getting a new release of EPICS from APS and pre-build the new EPICS release for SNS project. One should have both APS and ORNL cvs access privileges.
SNS local EPICS administrator is responsible for setting up local SNS ADE, getting a new EPICS release from the SNS central repository and building it in the local SNS ADE. One should have ORNL cvs access privilege.
SNS EPICS application developer is responsible for testing, modifying and extending one or more EPICS applications. One should have ORNL cvs access privilege.

Types of SNS software products

IOC application products are the control software products that run on an IOC and usually are the low-level control system applications, such as *RF*, *Vac*, *Magnet_ps*, etc.
Support module products are the control software products that run on an IOC and usually are meant for use by other IOC application products, such as *GPIB*, *IPAC*, *ControlNet*, etc.
EPICS extension products are the products that run on a Unix machine and usually are the EPICS Channel Access client applications, such as *MEDM, BURT, ALH, ar* etc; or the tools for supporting EPICS control system related products, such as *SDDS* for supporting high-level control system applications; or *makeBaseApp* for supporting building low-level control system applications.
Other products are the products that are control system related, but not part of EPICS tool kit, software products and usually are the commercial products, such as *Capfast*, *LabView*, etc. or non-commercial products, such as *CDEV*, etc.

II. Overview of SNS/ADE CVS Repository Structure

The SNS/ADE CVS repository is currently setup on sofa.ic.ornl.gov at ORNL. The structure is described in the following.

$CVSROOT = /cs/epics/cvsroot

$CVSROOT/

- support/
  - extensions/
  - base/
    - *** base-R3.13.1.1 is currently direct under base dir ;
      future APS import should start from a new sub-directory ***
    - base-R3.13.3/ ← Note: each new import from APS should start from a new sub-dir
  - front_end/
    - ab/
    - group3/
    - scope/
    - …
  - linac/
  - ring/
    - cnet/
    - mpf/
    - …
  - instrumentation/
  - target/
- iocapps/
  - utility/ ← any thing goes here
    - demo/
    - example/
    - test/
    - …
  - front_end/
  - linac/
  - ring/
    - vacuum/
    - tune-meter/
    - …

```
instrumentation/
target/
    ttf/          ←----- this is <app1>
        Makefile
        config/
        host/       ← where you put medm, alarm config files, etc.
        ttf<xxx>App/              ←----- this is sub-app <xxx> for <app1>
            Makefile
            <aaa>Db/
            <bbb>Db/
            src/
            host/
                adl/
                ahl/
                burt/
                . . .
        ttf<yyy>App/              ←----- this is 2nd sub-app <yyy> for <app1>
        . . .
        iocBoot/
            Makefile
            nfsCommands
            iocttf1/              ←----- this is 1st ioc  for <app1>
                Makefile
                cdCommands
                st.cmd
            iocttf2/              ←----- this is 2nd  ioc  for <app1>
            . . .
        OPI/                      ← where the dm2k files, etc.
        DOC/
                Readme.pdf        ←----- the overall release document for  <app1>
    <app2>/             ←----- this is another application for target subsection
```

III. SNS/ADE Development Directory Structure

To be able to share same control software ADE among the SNS collaborating labs and centralize SNS software product control and release to minimize future maintenance overhead, it is required that the following development directory structure will be setup identically on the machines at the each SNS collaborating labs. Tools to use to build this tree will be described in <<Procedures>> section in this document.

```
<VxWorks> = /cs/vxworks/              ← This is the VxWorks installation directory
                setup-tornado101      ← script to set up the path and environment variables for
                                          tornado 101; it is called by ADEsetup script.
                tornado101/
                setup-tornado20
                tornado20/
                . . .
```

<ADE> = /cs/epics/ADE

<ADE> ------- **iocs/**
|
---- **R3.13.1.1/**
|
---- **R3.13.2/**
|
---- **R3.13.3/**
|
---- ADEsetup-R3.13.1.1
ADEsetup-R3.13.3
ADEsetup -> (point to latest working release)

<ADE>/iocs/ ----- vxworks/
|
--- all/
|
--- <iocsns1>
|
--- <iocsns2>
|
---<iocsnsN>

<ADE>/iocs/vxworks/
tornado101/
mv2300-603/ ← *the images in this directory are copies from <VxWorks>/<tornado-version>/target/config/mv2300/*
vxWorks
vxWorks.sym
mv2300-604/
mv2100/
…

tornado20/
…

<ADE >/iocs/all/ ← *where you put the files for all iocs to use (common files)*
netInit.cmd ← this file is called by the startup.cmd of all the iocs

<ADE>/iocs/<iocsns1>/
startup.cmd
vxworks -> ../vxworks/<vxw-version>/<arch>
appioc -> /cs/epics/ADE/<epics-ver>/iocapps/<area>/<app>/<app-ver>/iocBoot/ioc<app>

<ADE>/iocs/<iocsns2>/

<ADE>/R3.13.1.1/
support/
R3.13.1.1-SNS1/
R3.13.1.1-SNS2/
base -> R3.13.1.1-SNS2 (base points to most recent SNS revision for R3.13.1.1 release)

*** NOTE: Unlike in CVS repository, all support products directly under <support> and they are shared SNS-site wide in <work-area>

ab/
1-1/
2-1/
group3/
mpf/
1-1/
cnet/
0-4/
0-5/
.

iocapps/
utility/
front_end/
linac/
ring/
vacuum/
1-0/
instrumentation/
target/
ttf/
1-0/
Makefile
config/
host/
ttf<xxx>App/
bin/
<arch>
db/
dbd/
include/
Makefile
ttf<xxx>App/
Makefile
<aaa>Db/
<bbb>Db/
host/
src/
*.c
O.<arch>

```
        ttf<yyy>App/
        . . .
        iocBoot/
            Makefile
            nfsCommands
            iocttf1/
                Makefile
                cdCommands
                st.cmd
            iocttf2/
            . . .
          DOC/
                  Readme.pdf
    2-0/

<app2>
    1-0/

    2-0/
```

IV. SNS Control Software Product Control/Release Web Page

A web page will be designed and resided on ORNL SNS web server to enhance SNS control software product control and release management. To encourage to share same SNS ADE and use one centralized cvs repository, this web page will be designed only for use of browsing information, such as what are the software products in SNS cvs repository, the descriptions of the products, and how to get the products. There will be no uploading or downloading software products directly via this web page.

The following information for each SNS software product should be in the web page:
Part I. Products developed by SNS (*SNS home products*)

- The product name, such as *Ring_vac, ControlNet, etc*
- The release number or version number of the product and the release date
- SNS cvs tag number for the product
- Description of the product and how to build the product, such as makefile settings, etc. – Readme.pdf
- Original author's name
- SNS contact person for the product (e-mail address)

Part II. Products tested by SNS (*Other products*)

- The product name, such as *dm2k, ar, SDDS, etc*
- Where the product came from and the product test date
- The product original version number
- SNS cvs tag number for the product

- Description of the product and test results and how to build the product, such as makefile settings, etc. – Readme.pdf
- Original author's name
- SNS contact person for the product (e-mail address)

Part III. Products to be developed or tested by SNS
- The product name
- Description of the product – Readme.pdf
- The person who will develop or test the product
- SNS contact person for the product (e-mail address)

V. SNS Control Software Product Control/Release Procedures

Each SNS control software product has to be checked into SNS centralized cvs repository before its release. Push/Pull procedures for moving SNS software products between centralized cvs server at ORNL and the local development hosts at the SNS collaborating labs are described in the following.

Push Procedure (moving software product from local SNS development host to centralized SNS cvs server at ORNL)

1) A ***SNS application developer*** is assigned to be responsible for a SNS product, such as Ring-rf control (he is going to develop) or Archiver (he is going to evaluate and to be a SNS specialist for the tool)
   This information should be published in PartIII of SNS Control Software Product Control/Release Web Page so that duplicating effort can be avoided
2) When the product is ready to be release, a ***SNS local EPICS administrator*** will use remote CVS to upload the product onto SNS centralized CVS server via ssh
3) When the product is committed into SNS centralized CVS repository, a SNS central EPICS administrator will validate the product and its cvs tag and update the SNS Control Software Product Control/Release Web Page
   This information should be published in Part I or Part II of SNS Control Software Product Control/Release Web Page

Pull Procedure (moving software product from centralized cvs server at ORNL to local SNS development host)

1) A ***SNS application developer*** requests to download a SNS software product to local development host after he read the SNS Control Software Product Control/Release Web Page
2) A ***SNS local EPICS administrator*** uses remote cvs check out the product from SNS centralized cvs host at ORNL to local development host via ssh

VI. Using CVS for the "Push or Pull" Procedures in SNS ADE

In this section, the "Push or Pull" procedures are detailed at the cvs command level. We suggest that you should walk through the example and exercise the procedures at the cvs command level. It assumes that the SNS ADE has been set up on your machine. We use sofa.ic.ornl.gov (where the SNS central CVS repository is located) at ORNL and sun1.sns.bnl.gov at BNL as example working hosts to demonstrate how the cvs is used in SNS ADE locally and remotely.

A. Create a new SNS application product in CVS repository

 # if you have already done the first 3 steps, skip to step 4.

1. cd /cs/epics/ADE
2. cp ADEsetup-template .ADEsetup-<uid>
   #Example: cp ADEsetup-template .ADEsetup-tang
3. Edit your ADE setup file
   # the above 3 steps are only applied to first SNS ADE users
4. source .ADEsetup-<uid>
5. cd $APP/<area>
   # where <area> could be ring, target, etc Example: cd $APP/ring
6. mkdir -p <myapp>/working; cd <myapp>/working
   # Example: mkdir -p vacuum/working; cd vacuum/working
7. $makeBaseApp -t example <myapp>
8. $makeBaseApp -i –t example <myioc>
9. cvs import -m "New application, myapp, initial import to cvs" iocapps/<area>/<myapp> <myapp> release-1-0
   # Create directory structure in CVS repository for your application
10. cd ..
11. cvs checkout -d 1-0 iocapps/<area>/<myapp>
12. rm -rf working

B. Create a module for your application in CVS repository

 # you only need to do this procedure once per application

1. source .ADEsetup-<uid>
2. cvs checkout CVSROOT
3. cd CVSROOT; Edit CVSROOT/modules file
   # define your unique key for your application in the modules file; Example: ring-vacuum iocapps/ring/vacuum
4. cvs commit -m "Added a new module ring-vacuum" modules

C. "Push" an application product to SNS CVS repository

1. source .ADEsetup-<uid>
2. cd $APP/<area>/<myapp>/<release>
   # cd $APP/ring/vacuum/1-0
3. cvs commit -m "Ring vacuum application release 1-0 is completed" .

4. cvs tag <tag-name> .
   # Example: cvs tag ring-vacuum-1-0
5. update “SNS Control Software Product List” web page

D. “Pull” an application product from SNS CVS repository

1. source .ADEsetup-<uid>
2. cd <working-area>
3. cvs checkout -r <app-tag-name> <app-module-name>
   # Example: cvs checkout -r ring-vacuum-1-0 ring-vacuum