

DRAFT: 15 January 1999

SNS EPICS Control System Application Development Environment (ADE)

Working Group Members:

Steve Lewis (LBL)

John Munro (ORNL)

Johnny Tang (BNL)

Rozelle Wright (LANL)

Introduction

This document summarizes results of the SNS EPICS control system Application Development Environment Working Group (ADEWG) which met on 8 January 1999 at Los Alamos.

A major challenge for the SNS control system development effort is configuration management and version control since initial application development efforts will be carried out at five different DOE laboratories, with integration of hardware occurring in Oak Ridge over a three year period. To meet this challenge, the Integrated Controls Working Group (ICWG) will attempt to keep all parts of the EPICS development environment integrated throughout the course of the SNS construction project.

Objectives

Objectives proposed by Steve Lewis and agreed to by members of the working group are:

1. Track development of the ANL/APS versions of EPICS during SNS development/construction.
2. Avoid the need for an integration step across SNS sites.
3. Maintain a single version of rules for building EPICS and EPICS-based applications for the whole SNS project.
4. Designate a single individual with responsibility to maintain the master version of EPICS used on the SNS project. This person does not have to be located in Oak Ridge.
5. Set up and maintain separate "development" and "operations/production" directories for each SNS EPICS release.
6. Maintain the SNS EPICS software releases on a single CVS server, located in Oak Ridge.
7. Minimize the amount of "true" EPICS knowledge required, i.e., try to eliminate the need for installation procedure guides such as the "SAFE" document prepared by Steve Lewis.

Proposed Application Development Environment

File and directory structures that constitute the ADE are laid out roughly according to the following hierarchy, starting at the top level:

1. EPICS release used for SNS
2. For each release set up directories by administrative categories, e.g., master, devel, repository, target, host
3. Application area
4. Specific applications
5. Major directories for each application (bin, src, db, dl, sch)
6. Executables, scripts, readme, makefiles, configuration files, template files, i.e., the nitty-gritty details

ADE files may be regarded as extensions of Unix into the control system domain of EPICS: the same organizing principles and conventions apply as are used for building and maintaining Unix system files.

Reasons for defining an SNS ADE include

1. Simplifying Makefile rules used to build an application area.
2. Extending the degree of automation in building new applications or new versions of existing applications.
3. Promoting an integrated development effort across lines of SNS major subsystems (front end, linac, storage ring, target, instrumentation).
4. Supporting use of the CVS server for effective version control.
5. Rationalize terminology used to refer to directories using a system of generic labels.

Implementation Path

Administrative

1. Designate an EPICS configuration/version control manager for each SNS DOE laboratory with the manager at ORNL also serving as the master coordinator for SNS EPICS configuration control.
2. Each configuration/version control manager will be given access to the SNS EPICS CVS server operated at ORNL.
3. The EPICS configuration/version control managers will coordinate development activities at each laboratory in a manner that will preserve an integrated EPICS environment.

Technical

Proposed Scenario

Requirements for implementation can best be identified by constructing a scenario that describes how a particular application will be developed. A likely scenario is expected to proceed as follows (from the very beginning):

1. Get files from the SNS master site CVS server and build the EPICS base.
2. Set up <epics-sns-nn>.
3. Execute Makefile to create a new **application area**.
4. Execute Makefile to create a new **application**.
5. Develop a new application.
6. Execute Makefile to create a new version of an application.

Discussion of Scenario

Building the EPICS base will require modifying .setup and configuration files.

Syntax examples for executing the various Makefiles will be as follows:

1. For a new **application area**
mkNewAppArea <top-name>
2. For a new **application**
mkNewApp <app-name> [<ioc-name>]
3. For a new version of an application
appNewVer <app.name> <new-version-number>
4. For a new IOC
mkNewIOC <ioc-name> <Vx-version> <ioc-type>
to go with a directory structure that looks like

```
<ioc-name>
  vx -> vxarea/ver/...
  vx/
  epicsV ->
  st.cmd
  nvram.boot
  channel_access_security_file
```

A setup script will be used to define the environment variables. A suggested syntax for execution of this script is

```
setup <epics-release-number>
```

with the following **list of environment variables** to be defined:

```
EPICS = <EPICS-root-path>
base = <base-root-path>
APPSarea = <APPSarea-root-path>
IOCare = <IOCare-root-path>
Vx_base = <Vx_base-root-path>
install = <operations-version-root-path>
```

Another environment variables to define might be “test” for a test version area.

A number of file and directory structures were proposed during the day and a half of discussion. No conclusions were reached regarding what the best layout would be, so the alternatives will be given below.

For organization of directories on the CVS server, the following was proposed (directories are indicated as “dir/” and files as “file”):

```
<CVS-root>/EPICS/
  APPS/
  IOCS/
    <ioc1>/cmd/
    ...
    <ioc2>/cmd/
```

Variants of file and directory structures for application areas which were discussed are presented below. They are not intended to be mutually exclusive, but were constructed in attempts to solve particular problems. The final form given the ADE must therefore draw from each of these.

VARIANT 1:

```
<root>/epics/SNS1/
  base/
    src/
      rec/
      dev/
      drv/
      config/
    config/
  .setup
  setup/
  SNS2/
```

VARIANT 2:

```
<top>/
  <app-name>/
    <version-1>/
      makefiles/
      CVS/
      sch/
      cmd/
      src/
      readme.appnew
    <version-2>/
```

VARIANT 3:

```
<app-area>/config/
  <...>.host
  rules.Vx
  xxxApp/
    <ver1>/
      sth
      Db/
      src/
        *.dbd/
        *.c
        *.cc
        *.?
        makefile.host
        makefile.Vx
        O.<...>
        o.<...>
        sharelist
      OPI/
      iocList/
        <ioc1>/
        <ioc2>/
        ...
        <iocN>/
    <ver2>/
    ...
    <verN>/
  bpmApp/
  magApp/
  ABapp/
  myAiApp/
    <1-0>/
    <2-0>/
    ...
    <n-0>/
```

Discussion

As the variants above show, a standardized approach to version control has not been reached. The issues revolve around deciding what level(s) in the hierarchy should be selected for the most effective management of version control. This needs to be done in a way to provide a reliable way to track changes

where most changes are occurring. It also needs to be done in a way to permit writing and implementing a clear set of rules in the makefiles and configuration scripts. There is a trade-off to be made between use of makefile rules and configuration template files to provide a desirable level of flexibility in making modifications without introducing elements that interfere with maintaining coherence in the development of new versions and releases of the software.

Variant 3 introduces a new feature (located at “<app-area>/xxxApp/<ver1>/src/sharelist”) designed to summarize in one place all the choices made to configuration and script files to build the specific application.

Scripts for setting up IOC applications could be gathered into a structure as follows:

```
<ioc-number>/st.cmd/  
    ldepics  
    idApps/  
        myAiApps/  
        bpmApp/  
        fsdApp/  
    epicsGo  
    mit                /* to start sequencers
```

Documentation for the current distribution of EPICS (R3.13.b12) refers to directories with such generic labels as <top> and <share> which make identification of the context difficult. Use of generic labels needs to be modified so that it is easier to identify the context in the documentation, e.g., use “target”, “host”, “Vx”, and “Unix” as filename extensions for script and template files to distinguish the different contexts.

Efforts will be made through the wider EPICS collaboration to seek adoption of an SNS ADE and its incorporation into the official EPICS software distribution in order to simplify configuration management for SNS.

Next Steps

1. Present this memo to Marty Kraimer (ANL/APS) at the next ADEWG meeting, tentatively scheduled for late February 1999 at ANL.
2. Discuss the variants and trade-offs, then try to reach consensus on what approach to take.
3. Define the ADE for the next release of EPICS (the starting release for SNS EPICS applications).

Conclusion

Modifications and new features to the EPICS ADE are proposed that are intended to support more reliable management of the EPICS software for use on the SNS project.