Displays and Scripts Release Development Environment
Short Term Transition Plan

21-May-2003

This document outlines the transition plan for modifying the development environment for displays and scripts. We are doing displays and scripts only at first, in order to get working examples in place from which other types of software (c/c++ code, python, matlab etc) can be added by following the pattern established by these.

To implement the transition involves changing a couple of very minor things in the existing development filesystem (a symlink and some PATH values), which we think will not be invasive, but we need to check with the group before proceeding.

1 Skeleton Infrastructure Transition Plan

The short term goal is that we want to create the skeleton of the release escalation mechanics. The basic release support involves the following subsystems, which are co-dependent. Our problem right now is to move them from their testing stage, to implementation, respecting that interdependency and the fact that we're implementing into a live control system.

1. **Release Directories**: the directories that implement the release escalation (ie dev, new, prod etc on dev and prod hosts). **Status**: Some of the release directories have been created; others await verification of this transition plan from the group. In particular there may be issues caused by breaking the symlink on DEV which makes $CD_SOFT/prod point to $CD_SOFT/dev.
2. **Environment variables**: The unix environment variables that point to the release directories (eg CD_DEV_BIN, CD_PROD_DISP etc). **Status:** The edits have been made to setEnv.csh, but setEnv.csh and the other scripts which implement the environment have not been released until the group is made aware of changes these scripts will make to the PATH environment variable on dev, see below.
3. **Path manipulation**: to help a user say which level of release (in their working directory, tst, dev, new, prod), from which they want to use a product. **Status**: the required environment variables definition scripts, and the path manipulation script have all been written, but not implemented until group has been warned of possible effects, see below.
4. **Distribution support scheme**: that copies files from dev to prod. **Status**: is being tested, but isn't going to be released until the directories that it copies files back and forth between exist (see 1 above).
5. **Makefile scheme**: that builds a product like a display and moves it from level to level (tst, dev, new, prod). **Status:** makefiles for displays and scripts are in development – see below.

We hope to implement the new support for each of these in the order, 1 to 5, above.

1.1 Scope

When the above 5 have been done, we can then start to move displays, scripts (executable and non-executable) and such into this release mechanism, and implement the security features, such as write-protecting the reference area etc.

1.2 Dependencies

To implement the five subsystems above we need the group to think about the possible effects of two prerequisites of their implementation:

1.2.1 Symlink of prod to dev on DEV

There presently exists a symlink between $CD_SOFT/prod -> $CD_SOFT/dev on development machines. We need to break that symlink so that we can create the release directories under

$CD_SOFT/prod on development. If any file, or link, in any of the directories below, or a subdirectory of these directories, are being used by accessing them using a filepath that uses the prod symlink, then when the symlink from CD_SOFT/prod -> $CD_SOFT/dev is broken, that file will no longer be accessible.

Presently on development, $CD_SOFT/dev/ (and therefore $CD_SOFT/prod) contains
1. displays in $CD_SOFT/dev/disp
2. scripts in $CD_SOFT/dev/script
3. script in $CD_SOFT/dev/@sys/bin
4. matlab files in $CD_SOFT/dev/matlab

So, is there any file or link accessed something like /afs/slac/g/cd/soft/prod/script/myscript, or anything else with prod in the filepath?

If there is, please:
1. Let me know, so I can coordinate when to break the link.
2. Change the filepath with which that file is accessed so that it does not use the prod symlink.

This question is not an issue for the gateways because, as yet, there is no prod directory under $CD_SOFT on the gateways.

1.2.2 PATH

The release directories have to go into the PATH on dev and prod. Specifically, after 3 above (Path Manipulation) is implemented, all user logins or processes that source ENVS.csh will include (among others) the following directories, which presently do contain executables, in their PATH:

On development
Until the symlink from prod to dev is broken, these are the same dir!
/afs/slac/g/cd/soft/dev/@sys/bin
/afs/slac/g/cd/soft/prod/@sys/bin

On production (gateways):
/afs/slac/g/cd/soft/prod/@sys/bin

ENVS.csh (actually a script ENVS.csh winds up calling, setPath.csh) will put these directories towards the head of the PATH (after "." if dot is present, but otherwise at the head).

Possible problem: Since these directories presently contain scripts, putting these directories in the "default" path of a programmer's or a process' login, may cause unexpected results. Scripts not previously in a process' PATH will now be available for execution. Also, since they're at the head of the PATH, even if there are other locations for scripts with the same name, those in the directories above will get executed in preference to the others.

Action Item: Need everyone to look at existing scripts in release directories to verify whether they should be in the default PATH of a process on that machine (dev or prod):
On afs (development) check scripts in /afs/slac/g/cd/soft/dev/@sys/bin/
On nfs (production) check scripts in /usr/local/cd/soft/solaris/bin/

1.2.3 EPICS_DISPLAY_PATH

Just as the problem above for PATH, when the support for 3 above (Path Manipulation) is implemented, all user logins or processes that source ENVS.csh will include (among others) the following directory, which now contains displays, in their EPICS_DISPLAY_PATH on DEV or PROD:

On Development:
`/afs/slac/grp/cd/soft/dev/disp/`

On Production (when cddev sources ENVS.csh):
`/usr/local/cd/soft/dev/disp/`

ENVS.csh (actually a script ENVS.csh winds up calling, setPath.csh) will put these directories towards the head of the EPICS_DISPLAY_PATH (after "." if dot is present, but otherwise at the head).

Possible problem: Since these directories presently contain displays now, putting these directories in the "default" path of a programmer's or a process' login, may cause unexpected results. Displays not previously in a process' EPICS_DISPLAY_PATH will now be available found and executed. Also, since they're at the head of the path, even if there are other locations for displays with the same name, those in the directories above will get executed in preference to the others.

Action Item: Need Judy and Kristi to verify that it's ok to put /afs/slac/grp/cd/soft/dev/disp/ in the EPICS_DISPLAY_PATH of a programmer on development (this should be a no brainer), and that it's ok to put /usr/local/cd/soft/dev/disp/ in cddev's EPICS_DISPLAY_PATH on production right now.

2 Transition Plan for Skeleton Infrastructure Job List

2.1 Release Directories

1. Everyone looks at display startup files, scripts (executable or non-execuatble) and anything else, like matlab, on development, and checks whether any use a filepath with prod in their filename.
    a. Fix any instances we find.
2. Break the symlink from prod to dev (Greg)
3. Create the remaining directories per the email of 5/12/03, attached as Appendix A. (Greg)

2.2 Environment Variables

4. Everyone look at existing scripts in the planned release directories (see 1.2.2) to verify whether they can be in the default PATH of a process on that machine (dev or prod). If not, tell Greg.
5. Put scripts now in ~brobeck/SETUPS/ in CVS in cvs/common/script. (Ken)
6. Get everyone to put ENVS.csh into their .cshrc (or into HEPIX?). (Greg and Jingchen)
    a. Can we put ENVS.csh into ccdev's cshrc on production at this stage Kristi?

2.3 Path Manipulation

Ken has written a utility, named setPath.csh, which can help you easily change the various important paths, PATH, EPICS_LIBARAY_PATH, LD_LIBRARY_PATH etc, to add or remove the various release directories or your working directory. This is completely analogous to TESTSHRX.

7. Add "alias setpath 'source $CD_SCRIPT/setPath.csh' to setAlias.csh, to make running setPath a little easier. (Greg)

2.4 Distribution Scheme

The distribution scheme needs 2.1 and 2.2 above to be in place before it can be implemented.

8. Add ssh keys for all developers on gateway4. (Jingchen)

2.5 Makefiles

The makefile support needs 2.1 and 2.2 above to be in place before it can be implemented.

3 Transition Plan for Scripts and Displays Job List

Following the implementation of the skeleton infrastructure described above, and adequate testing, we can then move the support for scripts and displays into it.

3.1 Scripts

The status of support for scripts is that the makefile scheme for moving scripts from their place in the install directory and up through the release directories is not completed. There is a bug in "blocking". That is, presently, and script in a lower level of release would be escalated to a higher level if any script wheich was already at a higher level were escalated. Hope to fix that soon. Following this fix, makefile support for scripts can be implemented according to the following transition plan:

1. Rename common/script to common/setup (Greg)
2. cvs import all scripts on development in /afs/slac/g/cd/soft/dev/@sys/bin bin (that are not already in CVS), into CVS. Need further work to determine which subdirectories in CVS these should go in. (Greg)
3. cvs import all scripts on production in /usr/local/cd/soft/solaris/bin (that are not already in CVS) into CVS . (Greg)
4. Write manifest files for scripts in common/setup (Greg)
5. Write manifest files for scripts from other sources (Kristi)

3.2 Displays

3.2.1 Transition Plan for Displays

Add ENVS.csh to cmdSrv scripts starting displays. This will cause the EPICS_DISPLAY_PATH to include the disp/ release directories.

The same "bug" that Greg reports in the Scripts section above about escalation beyond DEV also applies to the Displays. But, we can go ahead and release a more automated procedure that helps the display developers get their displays into production more easily. Here's what's left to do:

1. Change the existing display Makefiles to use the latest scheme. (Mike)
2. Make sure the environment variable EPICS_DISPLAY_PATH is setup correctly. (Mike)
3. Make sure $CD_SOFT/ref/disp/config is consistent with $CD_SOFT/cvs/disp/config. (Mike)
4. When no user has disp/config checked out of CVS remove $CD_SOFT/cvs/disp/config and $CD_SOFT/ref/disp/config and re-import the displays into CVS in $CD_SOFT/cvs/gui/disp/config and generate the reference area $CD_SOFT/ref/gui/disp/config. (Mike)
5. Perform initial display gmake to populate the $CD_SOFT/ref/disp and $CD_Soft/dev/disp areas.
6. Update web documentation with the latest Display development procedure. (Mike with help from Greg because Mike hasn't been following the new web page scheme).
7. Fix the Display README files to contain ????. (Mike – Kristi didn't like me having the display release procedure in the README files)
8. Write rdist manifest files. (Mike)

4 Documentation Job List

Need to write the following documentation soon:

4.1 Users' Guides

Documentation for helping developers develop their projects:

1. Help for using the makefile and release escalation system for displays (Mike and Greg)
2. Help for using the makefile and release escalation system for scripts (Greg)
3. Help for creating a new directory of scripts or displays. (How to create a dir under common/, get template Makefile and Makefile.Host from CD_REF_COM/template etc.) (Greg)

4.2 Programmers' Guides

Documentation for helping developers extend the functionality of the development environment itself.

1. Update Overall Unix Development Environment Requirements and Design per change to installing software in ref at the first stage of release. (Greg)
2. Write programmers guide to the development environment setup, so programmers can add to or modify the ENVS.csh, pathSetup.csh etc setup files. (Greg)
3. Write documentation standard for programmers and users' guides. This should cover such things as the fonts and styles used in examples. (Greg)
4. Guide to the deployment support (ssh keys, RDIST and so on). (Jingchen).
5. Guide to the deployment facility itself. (James).
6. Help for modifying a makefile for adding a new kind of software product, like "matlab". (Greg)

APPENDIX A

SHORT-TERM PLAN FOR CREATING RELEASE DIRECTORIES ON DEV AND PROD

```
RELEASE DIRECTORIES REQUIRED ON DEVELOPMENT
===========================================

Those directories marked "*" need to be created:

a. Displays:
------------
    ref/disp        - already exists, because it is presently the install dir
*   ref/dm-disp@ -> ref/disp
    dev/disp        - already exists. There should not be a dev/dm-disp@ link
                      however, as there exists now, because that link was
                      only used to help the makefile disambiguate dm from
                      dm2k display builds, and onlce they're built, as they
                      will be by the time of being moved to dev, they're
                      already built. Hence no need for dm-disp link.
*    new/disp
*    prod/disp
*    bck/disp

b. Scripts (non-exectable):
---------------------------
*    ref/script
     dev/script        already exists, and since non-executable scripts
                       aren't escalated past dev/script there will not be a
                       new/script, prod/script nor bck/script.

c. binary and executable scripts:
---------------------------------
*    ref/solaris@ -> sun4-solaris2
*    ref/sun4x_57@ -> sun4-solaris2
*    ref/sun4x_58@ -> sun4-solaris2
*    ref/sun4-solaris2
*                      /bin - won't do /sbin until security upgrades next week
*                      /lib
    dev/solaris@ -> sun4-solaris2
    dev/sun4x_57@ -> sun4-solaris2
    dev/sun4x_58@ -> sun4-solaris2
    dev/sun4-solaris2
                       /bin - won't do /sbin until security upgrades next week
                       /lib
*    new/solaris@ -> sun4-solaris2
*    new/sun4x_57@ -> sun4-solaris2
*    new/sun4x_58@ -> sun4-solaris2
*    new/sun4-solaris2
*                     /bin - won't do /sbin until security upgrades next week
*                     /lib
*    prod/solaris@ -> sun4-solaris2
*    prod/sun4x_57@ -> sun4-solaris2
*    prod/sun4x_58@ -> sun4-solaris2
*    prod/sun4-solaris2
*                     /bin - won't do /sbin until security upgrades next week
```

```
*                    /lib
*    bck/solaris@ -> sun4-solaris2
*    bck/sun4x_57@ -> sun4-solaris2
*    bck/sun4x_58@ -> sun4-solaris2
*    bck/sun4-solaris2
*                    /bin - won't do sbin until security upgrades next week
*                    /lib

ON PRODUCTION
=============
Most, but not all, of the above directories on development need a counterpart
on the gateways; in particular the "tst" directories (under ref) will not
have a production side equivalent.

Those directories marked "*" need to be created:

a. Displays
     dev/disp        already exists. There should not be a dev/dm-disp@ link
                     however, as there exists now, because that link was only
                     used to help the makefile disambiguate dm from dm2k
                     display builds, and once they're built, as there will be
                     by the time of being moved to dev, they're already
                     built. Hence no need for dm-disp link.
*    new/disp
*    prod/disp

b. Scripts (non-executable):
     dev/script      already exists, and since non-executable scripts aren't
                     escalated past dev/script there will not be a new/script
                     prod/script or bck/script on production either.

c. binary and executable scripts:

    dev/solaris@ -> sun4-solaris2
    dev/sun4x_57@ -> sun4-solaris2
    dev/sun4x_58@ -> sun4-solaris2
    dev/sun4-solaris2
                  /bin    - won't do /sbin until security upgrades next week
                  /lib
*   new/solaris@ -> sun4-solaris2
*   new/sun4x_57@ -> sun4-solaris2
*   new/sun4x_58@ -> sun4-solaris2
*   new/sun4-solaris2
*                 /bin    - won't do /sbin until security upgrades next week
*                 /lib
*   prod/solaris@ -> sun4-solaris2
*   prod/sun4x_57@ -> sun4-solaris2
*   prod/sun4x_58@ -> sun4-solaris2
*   prod/sun4-solaris2
*                 /bin    - won't do /sbin until security upgrades next wee*
*                 /lib
```