

CVS

CVS Tagging and Branching How-To

```
# *****
```

Okay for FACET, we are using the following Target Architectures:

A. Host = linux-x86

Red Hat Enterprise Linux WS release 4 (Nahant Update 8) B. Target1 = RTEMS-beatnik

RTEMS rtems-4.9.4, rtems_p0, sslApps_p3 C. Target2 = RTEMS-uC5282

RTEMS rtems-4.9.4, rtems_p0, sslApps_p3 # *****

Note: that now for modules used with EPICS R3-14-8-2 they will live here:

MODULES_SITE_TOP=/usr/local/facet/epics/modules/R3-14-8-2

Note: the modules for EPICS R3-14-12 will live here:

MODULES_SITE_TOP=/usr/local/facet/epics/modules/R3-14-12

CVS Tag Versioning convention:

<CVS_ModuleName>-RX-Y-Z-T,

where X is a number to indicate a major change where Y is a number where Z is a number Where T is a number where B is a number

::: Till/Murali please recommend our number scheme above.

```
#
```

```
=====
# Note: that now for modules used with EPICS R3-14-8-2 they will live here:
```

```
# MODULES_SITE_TOP=/usr/local/facet/epics/modules/R3-14-8-2
```

```
# Note: the modules for EPICS R3-14-12 will live here:
```

```
# MODULES_SITE_TOP=/usr/local/facet/epics/modules/R3-14-12
```

```
#
```

Okay, now we will make all the necessary MODS to build against EPICS R3.14.12

Any comments or feedback at this point?

Alrighty Then. 😊

(A) Modify, <TOP>/RELEASE_SITE, appropriately

(B) Modify, <TOP>/configure/RELEASE, appropriately

(C) Modify, <TOP>/configure/CONFIG_SITE, appropriately

Please look on facet-builder at these files for an example. 😊 Remember in this case TOP =

/usr/local/facet/epics/modules/R3-14-12/seq/seq-R2-0-11-lcls4

Since everything builds and works properly, I will of course commit the changes and tag this new release with a suffix of "lcls4" to indicate that this version works for both EPICS BASE R3-14-12 as well as R3-14-8-2 For full details always READ the RELEASE_NOTES!

Since we have not added any new features or bug fixes to this module what number do we increment to indicate that this module is linked against EPICS R3-14-12 and RTEMS 4.9.4? Hmmmm, what a dilemma. How about we use the minor number to indicate this?

Remember CVS Version Tag == RX-Y-Z, X= Major change; Y= Minor change; Z= Bug Fix

On the other hand, imagine that we are running Windows Vista and we switch to Windows 7:

I have source code for a super product muonFoobar; version muonFoobar-R1-2-1. Our customers

love it since there have been no bugs for 3 years!! 😊

If muonFoobar compiles and runs under both OS(s), I should not change the version number of muonFoobar as I have not added any new features or modified in source code.

In this case, I am just linking against the new libraries in the OS using the same API, etc...

So muonFoobar remains version muonFoobar-R1-2-1

Okay, so what do we do with this version of the sequencer which compiles and runs equally well on EPICS BASE 3.14.8.2 as well as EPICS BASE 3.14.12 ???

Well, we are going to tag it with "seq-R2-0-11-lcls4"

Be sure to make clean uninstall before tagging. 😊

```
cvs tag seq-R2-0-11-lcls4
```

(-- Last Step will be to release "seq-R2-0-11-lcls4" to production and lock it down. 😊

(-- RELEASE

```
cd $EPICS_MODULES_TOP
```

```
cvs co -r seq-R2-0-11-lcls4 -d seq-R2-0-11-lcls4
```

```
chmod -R g-w seq-R2-0-11-lcls4
```

(-- Don't forget to do your test/release plan for the Controls Deputy.

Okay, Guys and Gals we really should think about branch this RELEASE since it is now being used by our customers/users. This will also make it easier to bring in new modules from the EPICS community and merge on to our MAIN TRUNK without impacting the production branches.

```

#=====
# Why and When to Branch?
#=====
CVS allows you to isolate changes onto a separate line of development, known as a branch.
(1) For an official production RELEASE; we owe it to our customers to branch for major software components such as EPICS BASE, EXTENSIONS, and
MODULES. Not necessary to branch IOC Applications as they change much too often.
(2) Private developer branches should be created for "cutting edge"
development. These private branches are usually consider unstable.
(3) Since we get many of our modules from external sites; branching will make maintenance easy for modules that are released to production.
Modules from external sites must be imported on to the main trunk and merged. This can be a bit complicated and should not interfere with what we have
delivered(i.e. released) to operations.
# =====

# =====
# HowTo Branch if necessary:
# =====
A branch can be created to track and maintain any major development or change in the published API. We can also decide to create a branch when a
RELEASE is pushed to production.
Branching production RELEASES is recommended.
Branch_Name ==> <CVS_MODULE_NAME>-<PROD_RELEASE_TAG>-branch
Release Tags against this branch will look like the following:

Example, for the sequencer module:
<CVS_MODULE_NAME>_<VERSION-STRING>_branch
seq-R2-0-11-lcls4_branch
# =====

# =====
Steps followed to branch the "seq" EPICS Module:
# =====
(0) Log into the facet-builder EPICS linux-based file server

(1) source $EPICS_SETUP/go_epics_3-14-12_facet.bash

(2) cd $EPICS_MODULES_TOP

(3) pwd
/usr/local/facet/epics/modules/R3-14-12

(4) export CVSROOT=:ext:ernesto@lcls-prod02:/afs/slac/g/lcls/cvs

(5) cvs rtag -r existing-tag-name -b branch-tag-name project-name

(6) cd $EPICS_MODULES_TOP/seq

(7) Now create the actual Branch:
cvs rtag -r BASE_seq-R2-0-11-lcls4 -b seq-R2-0-11-lcls4_branch seq
Note: you can execute the above command from any directory.

(8) If a bug fix is needed Okay, let's checkout the branch:
cd
cvs co -r seq-R2-0-11-lcls4_branch -d seq-R2-0-11-lcls4_branch seq
(This is can be done with Dayle's new eco program as well. 😊 )
Dayle's program works very well!! 😊

(9) Let's go there and check if we are indeed on the branch by using
the "cvs status" command.
cd seq-R2-0-11-lcls4_branch
cvs status Makefile
=====
File: Makefile Status: Up-to-date

Working revision: 1.1.1.2
Repository revision: 1.1.1.2
/afs/slac/g/lcls/cvs/epics/site/src/seq/Makefile,v
Sticky Tag: seq-R2-0-11-lcls4_branch (branch: 1.1.1.2.4)
Sticky Date: (none)
Sticky Options: (none)
=====

Now feel free to fix your bugs or make very minor changes.

```