Developing along a GlastRelease Branch

Introduction

So you have a tagged released version of GlastRelease, and we desire to update this released version with tweaks, patches, and updated packages.

Updating the affected package(s)

First question to ask, has there been any updates to the HEAD of the package(s) that need updating? Can those changes be included or not?

If we can tag the HEAD and use it, then make your updates, tag the HEAD and proceed to the next step.

If not, then we must apply our changes to a branch (as well as the HEAD so we don't lose the updates in the future). Currently, all packages used by GR are automatically branched when GR is tagged. The branch naming convention is: GlastRelease-*v*i*r*j*p*k
Modifications to any packages should occur along this branch. See the workbook discussion concerning branches and how to access and tag them: http://glast-ground.slac.stanford.edu/workbook/pages/advanced_CVS/cvsBranches.htm

Performing the update to GlastRelease

Now that the package(s) have been updated and tagged, it is time to update GR itself. Checkout the GlastRelease-*v*i*r*j*p*k branch of GlastRelease:
cvs co -rGlastRelease-*v*i*r*j*p*k GlastRelease

Update the GlastRelease requirements file to use the new tag(s) by editing the requirements and make sure to update the doc/release.notes as well. Now check in your changes

cvs ci -m"log message"

When the check-in is performed you will notice the revision of the cmt/requirements file, note this string! You will use it to trigger a build of this HEAD version manually in the next step.

Triggering a build

The ReleaseManager currently ignores updates along the branches so no HEAD buid will start. However, we can trigger them manually, using one of Navid's ReleaseManager scripts. Assuming you are logged into a noric, you can do:

heather@noric06 $ ~glast/infraBin/ReleaseManager/trigger.pl
Usage trigger.pl [options]
option:
--tag The tag to use for the build. For example: rh9_gcc32
--package The package to build. For example: GlastRelease
--version The version of the package to build. For example HEAD1.123
--build The environement to build on. Default value of "" is linux. Other values are windows, mac

Note: for rhel4 builds, the --build option must be set to rhel4

It is recommended that a build be triggered for rhel4_gcc34, rh9_gcc32opt, rh9_gcc32 and VC8debug, at least. The rh9_gcc32opt version will be used by the system tests in the next step.

So to trigger a rh9_gcc32opt build we would do:

~glast/infraBin/ReleaseManager/trigger.pl --tag rh9_gcc32opt --package GlastRelease --version HEAD1.1027
Note we left off the --build option, as we are taking the default, linux.
The version is the revision that was reported when we did our cvs check-in in the previous step.

System Tests

It is recommended that systests be performed on the new GR branch HEAD before officially tagging the branch. Ask Leon or Liz!

Tagging Convention

Assuming the system tests went well, we are ready to tag the branch of GlastRelease.

The "p" portion of the tag will be reserved for the case where we are updating GR along a branch. By convention, patch tags start with 1 (one) and are incremented from there as necessary.

And now to tag.. we use the tag command applied to the working directory where you have checked out the GR branch. So assuming you are in the GlastRelease/v*i*r*j*p*k* directory, do:

cvs tag v*i*r*j*p*k*+1