Table of Contents

Preparing a software project for release

This wiki explains the procedure to be used by software project managers to prepare a project for release. It assumes the project is in CVS. For projects migrated to SVN, see the similar procedure here

Introduction

The procedure described below is a suggested procedure. Mandatory steps are highlighted in RED. In the code examples below, we assume you are releasing project `LBCOM` version `v99r11`.

Tagging

The project has to be correctly tagged in order to be ready for release. A project release consists of a set of constituent packages each of which is tagged with a CVS tag. The <Project>Sys and <PROJECT> CVS modules have to be tagged with the project <version>. The list of constituent packages and tags to be released is found in the `<Project>Sys/cmt/requirements` file.

The following is the recommended procedure for tagging the packages and project. Note that the procedure applies to projects. For data packages, it's enough to just tag the package, as in the last step.

1. Go to a working directory with lots of quota (typically a directory on a local disk of your machine)
2. Update the project dependencies and tag the `<PROJECT>` package
   - ♦ check out the HEAD of the complete project

     ```
     getpack -P -rh LBCOM HEAD
     cd LBCOM/LBCOM_HEAD
     ```
   - ♦ edit `cmt/project.cmt` to update the dependencies of the project
   - ♦ commit and tag with the new version number:

     ```
     cvs commit -m "dependencies for v99r11 release"
     cvs rtag LBCOM_v99r11 LBCOM
     ```
3. Tag all the packages that are updated for the release. The packages to be updated should be documented in the LHCbTagCollector.
   - ♦ Take the appropriate section of the tag collector page and, for each package mentioned in there:

     ```
     cd <theHat>/<thePackage>
     cvs diff -r <theLastUserTag>
     ```
   - ♦ If there are no diffs, all is well, the head of CVS corresponds to the tag that should be released. If there are diffs, you have to decide whether you want to include them in the release - either unilaterally or in consultation with the author of the changes.
   - ♦ In `doc/release.notes`, add the line with release version number and date
     - ◊ If already added by the package manager, check that the version number is correct, and that they haven't moved the line from a previous release - yes I happens!)
   - ♦ Check that `cmt/requirements` contains the new version number
     - ◊ If not, update it
   - ♦ commit and tag with the new version number:

     ```
     cvs commit -m <theNewOfficialTag>
     tag_package <theNewOfficialTag>
     ```
4. Update the `<Project>Sys` package

   ```
   cd LbcomSys
   ```

- in `cmt/requirements` update the version numbers of all the updated constituent packages
- in `cmt/requirements` update the `<Package>Sys` version number if not already done. The version must correspond to that used to tag the <PROJECT> package
- in `doc/release.notes` and add, for each of the modified constituent packages, a summary of the release notes of the package.
- Commit and tag

```
cvs commit -m v99r11
tag_package LbcomSys v99r11
```

Checks prior to release

1. Check that the whole project is correctly tagged
   - Go back to your root working directory and check out the tagged project as it will be released

     ```
     getpack --project -r LBCOM v99r11
     ```
   - If everything checks out correctly, you are almost done.
2. Check that nothing has been missed from the release

   ```
   cd LBCOM/LBCOM_v99r11
   cmpPackageWithTrunk --all
   ```

   This gives you any files that are in the head of CVS but not included in the release tag. This is either because you made a mistake in the tagging above, or because the changes were not included in the tag collector, either on purpose or by accident. In the latter case decide if the changes should anyway be included - either unilaterally or in consultation with the author of the changes. To add new tags:

   - go to the package to be updated, under `LBCOM/LBCOM_HEAD`
   - tag the missing package as above
   - update `LBCOM_HEAD/LbcomSys/cmt/requirements` and `LBCOM_HEAD/LbcomSys/doc/release.notes`
   - commit and retag (`cvs rtag -F v99r11 LbcomSys/cmt/requirements`)
   - go back to LBCOM/LBCOM_v99r11:

     ```
     getpack <theHat>/<theModifiedPackage> <theNewTag>
     getpack LbcomSys v99r11
     ```
   - check again that you now have all the wanted changes: `cmpPackageWithTrunk --all`

Nightly test of the software

The tagged software should be built and tested at least once in the LHCb nightly system. Ask someone with write access to the nightlies configuration file (Karol, Marco, Marco, Hubert, Gloria...) to add the tagged project to an appropriate slot of the nightlies (usually *lhcb-prerelease* slot)

Request @ Savannah

If the nightly tests are successful, you can ask for a release. The release request has to be made in the task tracker of the LHCb Deployment Savannah project.

Document the release

The doxygen documentation of the project is made automatically by the release procedure. But the project release pages (e.g. http://cern.ch/LHCb-release-area/DOC/lbcom/) have to be updated by hand, see instructions here

-- MarcoCattaneo - 23-Jul-2010

This topic: LHCb > PrepareProjectReleaseCVS
Topic revision: r1 - 2010-07-23 - MarcoCattaneo