Table of Contents

Preparing a release of DecFiles

Verify the status of the package

You will need to check the status of the package with respect to

- the previous version
- what is expected from the gitlab page https://gitlab.cern.ch/LHCb-SVN-mirrors/Gen-DecFiles/milestones and that the release notes reflect the status of the package.

In order to check the status of the package you can use the nightly build where the head (or a branch) of DecFiles is built: lhcb-decfilestests. For the final steps - or if you need to do any modification - you can use the gitlab web editor or get a local copy of the package.

Check there are no errors/problems with the newly committed event types

You can check if there are decay files with errors and for which the options have not been produced by looking at the error messages produced in the nightly. Syntax errors for which the options of the event types could not be built will appear as build errors. Simple tests for all event types for which decay files have been committed since the last release are automatically run in the slot: 5 events will be run and you can check if the jobs were successful. These checks can be done regularly and you can contact the people that committed the problematic decay files as soon as you spot a problem.

In most cases the errors messages will be of expected types, i.e. missing known keyword and you may fix them yourself without the need to contact the authors, while problems when running the events will likely need you to get in touch with them.

If you need to fix build errors in the decay files you will need to get a local copy of the package and build the options with the default settings of the script. It may be useful to dump in a file the messages as to be able to review them by

```
cd Gen/DecFiles
make >& make.log
```

Note that you will need to remove the event types produced to make them again otherwise the script will find them and refuse to make them again.

Checking the status with respect to the previous version are those expected from the tag collector

There are two scripts setup to help you do this. They are located in `$LHCBHOME/group/gauss/releases`. First do a diff of all subdirectories of the nightly (or your local copy) vs the previous release excluding the .svn directories. For the dkfiles and options directories save them in files, e.g.

```
diff -r --exclude ".svn" $LHCBRELEASES/DBASE/Gen/DecFiles/v27r14/cmt $LHCBNIGHTLIES/lhcb-branches
diff -r --exclude ".svn" $LHCBRELEASES/DBASE/Gen/DecFiles/v27r14/doc $LHCBNIGHTLIES/lhcb-branches
diff -r --exclude ".svn" $LHCBRELEASES/DBASE/Gen/DecFiles/v27r14/dkfiles $LHCBNIGHTLIES/lhcb-bran
diff -r --exclude ".svn" $LHCBRELEASES/DBASE/Gen/DecFiles/v27r14/options $LHCBNIGHTLIES/lhcb-bran
```

p.s. the 'Mon' above is given as example. Put the day for which you are doing the verification.

The differences in the various packages should be those expected (e.g. you know that the create_options.py has been modified as listed in the release notes and in the tag collector).

Now check the differences in the subdirectory you expect them. A script exist to list the real differences, new files or removed files in the dkfiles and options directory. For dkfiles do:

```
python $LHCBHOME/group/gauss/releases/DiffDecFiles-dk.py dkfiles-vs-v27r14.log
```

You will have first a report of the files with real differences: they should correspond to those that have been fixed in the metadata part for example. Then you will have a list of files only present in your local copy (the new one) and those only present in the version v27r14. Check this against the tag collector and the release notes. For the files only present in version v24r1 or for which the event type code has been changed, verify they are listed in the file doc/table_obsolete.py: if not check you will need to update it. See below.

You then have to check the difference between the new options produced and those in the latest version. A dedicated script exist also in this case:

```
python $LHCBHOME/group/gauss/releases/DiffDecFiles-opts.py options-vs-v27r14.log
```

Again you will see if people modified options (check if this make sense, i.e. does not change the meaning, for example the association eventtype-nickname or settings to produce exotic even types without this having been agreed on), new options and removed options. They should be consistent with what expected from the differences in dkfiles. Note that you can have more then one options produced from one single decay file.

Clean up obsolete decay files

Event types are unique identifier of MC samples and as such cannot be reused in a production campaign. Even if people remove or fix decay files the event type id most likely has to be kept, in particular if sample for it exist in the book-keeping or if releases in a DecFiles for a running production (now Sim08, i.e. >=v27r4). Check if decay files removed or with a modified event type have already been entered in the doc/table_obsolete.py, if not do so with a nickname that will indicate this is old/wrong/buggy. Don't forget to updated the release notes and the tag collector!

Tagging the release and final verifications

Get a local copy of the package

At this point if you have not already done so you will have to get a copy of the package as you will need to make some changes. In case it is for the latest development get the head revision.

```
git clone ssh://git@gitlab.cern.ch:7999/LHCb-SVN-mirrors/Gen-DecFiles.git Gen/DecFiles
```

Due to the special nature of DecFiles being a package you cannot use the tag collector utilities for releases.

Change version and put an header in the release notes

Make sure the version in the requirements file correspond to the one you are preparing the release for (e.g. v27r15) and put the header line with the version and date in the release notes Commit to git and tag with

```
git commit -a -m 'updated version number'
git tag v27r15
git push origin master tag v27r15
```

For a different branch (not master), use:

```
git tag v27r15 v26b
git push origin v26b tag v27r15
```

Build the tagged version and prepare the dev documentation

It is a good idea to build once the tagged version to check there are no lingering errors if somebody committed somethings after the last nightly build. You should do that in the `$LHCBDEV/GAUSS` area. **Note: Only Gloria and Patrick may be able to do so at the moment** Go to the area and clean up the previous version sitting there

```
cd $LHCBDEV/GAUSS/Gen/DecFiles
```

then get the tagged version you want to release and build it

```
git fetch origin
git checkout v27r15
```

and build it

```
make
```

You should not have any error/warning at this point. And you can now check that what you have is identical to what you had in the nightly before you added the release notes and tagged it.

To make the documentation go to `$LHCBDOC/decfiles`, change previous version in `make_df_doc.py` and use the script set up for it

```
cd $LHCBDOC/decfiles
python make_df_doc.py dev
```

Put the tagged version in the nightlies

You can now add the tag to the nightlies for a very final check with the qmtests (**Note: Only Gloria should do so at the moment since it is a integrated with the Gauss nightly. Ask her to put the version in the nightly.**) Go to the Nightly Builds Configuration Editor. Click on the lhcb-branches slot and on then on the arrow on the left: it will open up - Change the DecFiles version from HEAD to the one to be released.

Announce the availability of the pre-release for final checks

You should also send a mail to lhcb-mc-production@cernNOSPAMPLEASE.ch and when there are many changes to gauss@cernNOSPAMPLEASE.ch announcing that the pre-release of DecFiles vXrY (e.g. v27r15) is available and will be picked up by the Gauss nightly builds in the `lhcb-branches` slot and the list of decay files produced is on the web at http://lhcb-release-area.web.cern.ch/LHCb-release-area/DOC/decfiles/releases/dev/table_evttype.php.

Prepare the tag collector for the next release

Note: Only Gloria and Patrick have the privilege to do so at the moment. Ask one of them to do it, if not already done. Go to the tag collector and Click on New Project'. Enter 'DecFiles' as name of the project and the next version. Add the explanation on how to add a new change in the tag collector (you can cut and paste the one of the previous version). Once the new DecFiles (e.g. v27r16) is listed in the tag collector, edit the generic entry `GSDCTNXU, myDecayFile` and click on `Show/Hide` of `Extra associated project`. Click on new version of DecFiles and de-select the old version. The entry should now appear in the new DecFiles v27r16.

Request the deployment

After at least one successful use of the new version in the nightly you can ask for the deployement in the release area and on the grid. Go to savannah (https://sft.its.cern.ch/jira/browse/LHCBDEP) and once you logged-in submit a new task. Put the name and version in the subjects, choose the package and ask for a grid deployment. Also put Gloria.Corti@cernNOSPAMPLEASE.ch, Patrick.Robbe@cernNOSPAMPLEASE.ch, Tomas.Pilar@cernNOSPAMPLEASE.ch in the cc at the bottom. When asking for a deployment of two parallel releases, specify to run the bookkeeping update only for one (for example, for v26 but not for v25).

Update the documentation for the release and announce it

Once you receive the notification that DecFiles has been deployed on the grid, you should update the documentation: go to `$LHCBDOC/decfiles` and type

```
python make_df_doc.py v27r15
```

update the soft link releases/latest to point to v27r15 then edit `releases/v27r15/description.html` with some explanation (you can look at the examples for previous versions. Finally got into the Sim08 subdirectory and put a soft link to introduce this version in the Sim08 category and also update here the soft link to the latest version

```
ln -s ../v27r15 v27r15
ln -s ../v27r15 latest
```

Finally send an announcement to the lhcb-gauss@cernNOSPAMPLEASE.ch & lhcb-mc-production@cernNOSPAMPLEASE.ch mailing lists with a short explanation.

Set the version as released in the tag collector

The very last thing to do is to go to the tag collector and hide the DecFiles v27r15 version. When asked if you want to declare the project released answer yes.

-- GloriaCorti - 14-Nov-2011

This topic: LHCb > DecfilesReleasesGit
Topic revision: r1 - 2015-12-10 - MarcoClemencic