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 tag collector

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. The DecFiles package is build in the `lhcb_branches` slot together with `Gauss v45b` (for now - a separate test project will be set up separately in the future). For the final steps - or if you need to do any modification - you will need to get a local copy of the package.

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

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.

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 lb-clone-pkg  Gen/DecFiles
```

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 back to gitlab. Afterwards visit https://gitlab.cern.ch/lhcb-datapkg/Gen/DecFiles/tags and create new tag from master (or branch in case of special needs)

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. You should do that in the `$LHCBDEV/GAUSS` area.

```
cd $LHCBDEV/GAUSS/Gen/DecFiles
git pull
git checkout vXrY
make clean
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
```

Request the deployment

To ask for deployment, go to JIRA (https://its.cern.ch/jira/projects/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.

Update bookkeeping

From lxplus, once the changes on CVMFS have been propagated (check with `ls /cvmfs/lhcb.cern.ch/lib/lhcb/DBASE/Gen/DecFiles`), you have to update the bookkeeping.

- Get a valid Grid proxy

  ```
  lhcb-proxy-init
  ```
- Update the bookkeeping

  ```
  cd /cvmfs/lhcb.cern.ch/lib/lhcb/DBASE/Gen/DecFiles/<version>
  lb-run -c best LHCbDirac dirac-bookkeeping-eventtype-mgt-insert doc/table_event.sql
  lb-run -c best LHCbDirac dirac-bookkeeping-eventtype-mgt-update doc/table_event.sql
  lb-run -c best LHCbDirac dirac-bookkeeping-eventtype-mgt-update doc/table_obsolete.sql
  ```

The `dirac-bookkeeping-eventtype-mgt-insert` command will produce many errors like

```
{'EVTTYPEID': '12165431', 'DESCRIPTION': 'Bu_D02pi+pi-,K+pi-pi0=PHSP,DecProdCut', 'PRIMARY': '[B+
ORA-06512: at "LHCB_DIRACBOOKKEEPING.BOOKKEEPINGORACLEDB", line 1131
ORA-06512: at line 1
 )
```

This is normal as most event types already exist in the Oracle table. Things are OK if the last line is like

```
The following event types are inserted: ['42102201', '42102200', '42163203', '42102221', '4210222
```

Update the documentation for the release

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 Sim09 subdirectory and put a soft link to introduce this version in the Sim09 category and also update here the soft link to the latest version

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

Update production steps and models in Dirac

Announce new version

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

-- MichalKreps - 2018-02-27

This topic: LHCb > DecfilesReleases
Topic revision: r23 - 2018-06-05 - MichalKreps