# Table of Contents

Software Release Guide.........................................................................................................................1

Special Instructions and troubleshooting..................................................................................................2
  - Installation on AFS.................................................................................................................................2
  - Deployment from Old Tarfiles................................................................................................................2
  - Removing Software (archival)................................................................................................................2
    - Projects..............................................................................................................................................2
    - Data Packages.................................................................................................................................3
  - Troubleshooting (For Releasers)...........................................................................................................3
    - If there is a quota problem................................................................................................................3
    - In case of problems with incomplete installs...................................................................................3
  - Troubleshooting (For Librarians)...........................................................................................................4
    - If the nightly builds do not work.......................................................................................................4
    - If lb-release-rpm prints messages like "error: not an rpm package".................................................5
    - If everything has gone so wrong that you you want to sit in a corner and cry... or restart from scratch........................................................................................................................................5
  - For the Librarian....................................................................................................................................6
    - Releasing a new version of LCG externals.........................................................................................6
      - Until Gaudi v23r2.............................................................................................................................6
      - As from Gaudi v23r3..........................................................................................................................6
      - As from LCG 86..................................................................................................................................7
      - As from LCG 87..................................................................................................................................8
    - Updating the Software Configuration DB for a new LCG or Gaudi.................................................9
    - Releasing a new version of LCG Grid...............................................................................................9
    - Preparing the meta RPM with the software to be installed locally on the Online farm...............10
Software Release Guide

This page is moving to here
Special Instructions and troubleshooting

Installation on AFS

Since 2018-01-22, installation on AFS is not needed, but the old instructions are still available at ProjectReleaseOld.

Note that only librarians will be able to change the content of AFS, so contact <lhcb-software-librarians@cern.ch>.

Deployment from Old Tarfiles

Very old versions of software projects cannot be deployed with RPMs, in which case it's possible to refer to the instructions at ProjectReleaseOld.

Removing Software (archival)

Projects

- Archive the project documentation (step to be defined, for the time being try Python
  `$LHCBDOC/scripts/gc_archive.py <project> <version>`, noting that `<project>` must be in all lowercase). This must be done before removing the software!

- On `cvmfs-lhcb`, start a transaction in the usual way, then list the RPMs you want to remove using the `cvmfslibinstall` command:

```
cvmfslibinstall list URANIA_v6r2p1
URANIA_v6r2p1          1.0.0  1   local
URANIA_v6r2p1_index    1.0.0  1   local
URANIA_v6r2p1_x86_64_centos7_gcc62_dbg 1.0.0  1   local
URANIA_v6r2p1_x86_64_centos7_gcc62_do0 1.0.0  1   local
URANIA_v6r2p1_x86_64_centos7_gcc62_opt 1.0.0  1   local
URANIA_v6r2p1_x86_64_slc6_gcc49_dbg 1.0.0  1   local
URANIA_v6r2p1_x86_64_slc6_gcc49_do0 1.0.0  1   local
URANIA_v6r2p1_x86_64_slc6_gcc49_opt 1.0.0  1   local
URANIA_v6r2p1_x86_64_slc6_gcc62_dbg 1.0.0  1   local
URANIA_v6r2p1_x86_64_slc6_gcc62_do0 1.0.0  1   local
URANIA_v6r2p1_x86_64_slc6_gcc62_opt 1.0.0  1   local
```

Then you can remove the packages using `cvmfslibinstall remove`, passing it the list of files to be removed (it will order them correctly and check dependencies). Please note that lbinstall does **NOT remove directories if they contain files that do not belong to the RPM packages** (e.g. generated files). Therefore it is necessary to check the project top directory if still existing and check what is left and remove manually if necessary.

N.B. after removing the packages, the files created by hand that do not belong to any package are **NOT** removed. You may need to check the directory for those.

N.B. Be careful about the packages returned, `cvmfslibinstall list URANIA_v6r2` returns the packages for v6r2 and v6r2p1...

Remember to run `cvmfs_publish` at the end, as usual, to close the transaction.
Data Packages

- got to the CVMFS publishing machine (cvmfs-lhcb) and uninstall

```bash
cvmfs_transaction
cvmfslbinstall remove <PROJECT>_<Hat>_<Package>_<version>
cmvfs_publish
```

Troubleshooting (For Releasers)

If there is a quota problem

When copying tars. First check what's wrong

```bash
fs lq $LHCBTAR/*
```

Then increase quota with

```bash
afs_admin sq $LHCBTAR/<PROJECT> <quota>
```

In a project:

```bash
fs lq /afs/cern.ch/lhcb/software/releases/<PROJECT>/*
```

Then increase quota with

```bash
afs_admin sq /afs/cern.ch/lhcb/software/releases/<PROJECT>/<PROJECT>_<version>/*
```

In rpms... on 2/12/2016 Ben did (see JIRA):

```bash
for r in $LHCBTAR/rpm/lhcb/<PROJECT>_<version>_*rpm; do
  for p in `rpm -qp --requires $r | grep LCG`; do
    ./afslbinstall --disable-yum-check --just-db install $p;
  done;
done
```

In case of problems with incomplete installs

Sometimes the automatic installation may fail, for example because not all platforms were correctly built in the release build, or because some of the needed external dependencies are missing. Depending on the reason for the incomplete install, you may want to ignore the missing dependencies, and install anyway, or just install those platforms for which the dependencies are OK.

- First find the available packages:

```bash
/afs/cern.ch/lhcb/software/lbinstall/afslbinstall query "$\{MyProject^^\}$_\{MyVersion\}"
```

- There are two useful options of afslbininstall to help with partial deployment of the packages returned by the above query:

```bash
/afs/cern.ch/lhcb/software/lbinstall/afslbininstall install --nodeps <package_name>
```

installs just the package but none of its dependencies. Be careful with this because you have to deploy by hand, one package at a time, the whole dependency stack for the deployment to be useful.
install installs the complete set of dependencies, but skipping any missing dependencies

For completeness, afslbinstall called without options installs the complete set of dependencies but stops with an error if a dependency is missing. It is what is called internally by lb-deployment-afs-install

Hint: all the above can be condensed in a single command to install all platforms ignoring missing dependencies:

```
for r in `$(/afs/cern.ch/lhcb/software/lbinstall/afslbinstall query "${MyProject^^}_${MyVersion}" )`
  /afs/cern.ch/lhcb/software/lbinstall/afslbinstall install --no-strict $r
done
```

Or, if you want to exclude a platform (e.g. because it has missing dependencies):

```
for r in `$(/afs/cern.ch/lhcb/software/lbinstall/afslbinstall query "${MyProject^^}_${MyVersion}" | grep -v ThePlatformToExclude)`
  /afs/cern.ch/lhcb/software/lbinstall/afslbinstall install $r
done
```

Don't forget to do the same on cvmfs, where you replace
```
/afs/cern.ch/lhcb/software/lbinstall/afslbinstall with cvmfslbinstall
```

### Troubleshooting (For Librarians)

#### If the nightly builds do not work

The nightly build system is nothing more than wrappers around simple tools that can be called by any user, so it is possible to prepare the RPMs for the release by hand.

- Go to an Centos7 machine with AFS (e.g. lxplus or a build machine), to a temporary working directory (e.g. /build/$USER)

- Prepare the configuration file to drive the build
  
  `lbn-gen-release-config MyProject vXrY > lhcb-release.json`
  
  or
  
  `lbn-gen-release-config --cmt --pack 'Hat/Package vXrY' > lhcb-release.json`
  
  for data packages

- Checkout the sources and prepare the shared RPM

  `lbn-checkout --verbose lhcb-release.json`
  
  `lbn-rpm --shared --builddir tmp/checkout --verbose lhcb-release.json`

- For each required platform build, test and prepare the RPM (not needed for data packages)

  `export CMTCONFIG=...
  lbn-build --verbose --clean --with-tests -j 4 lhcb-release.json`
  
  `lbn-rpm --verbose lhcb-release.json`

In case of problems with incomplete installs
• Check if the builds/tests are ok (not needed for data packages)
  ♦ start the Python mini web server

    cd artifacts && python -m SimpleHTTPServer
  ♦ with a web browser connect to the port 8000 of the build machine used (e.g. http://lxbuild00.cern.ch:8000/)
  ◊ build log: summaries.<CMTCONFIG>/<Project>/build_log.html
  ◊ test reports: summaries.<CMTCONFIG>/<Project>/html

• Publish the RPMs (from lxplus.cern.ch)

    cp -v -n artifacts/MYPROJECT*.rpm $LHCBTAR/rpm/lhcb

then, from lxplus.cern.ch

    createrepo --workers=20 --update $LHCBTAR/rpm/lhcb

• If lxplus.cern.ch works, continue with the procedure to install to AFS, otherwise you should install on AFS from the generated tar files, then continue with the normal procedure

If lb-release-rpm prints messages like "error: not an rpm package"

It might mean that the RPM file is probably corrupted.

This could happen during the copy, in which case it is enough to remove the corrupted files from $LHCBTAR/rpm/lhcb and call again lb-release-rpm.

If it still does not work, check that there are no permanent problems in $LHCBTAR/rpm/lhcb (e.g. quota) and that the files are copied correctly with something like (bash)

    for f in /eos/project/l/lhcbwebsites/www/lhcb-nightlies-artifacts/release/lhcb-release/${build_id}/*.rpm; do cmp $f $LHCBTAR/rpm/lhcb/$(basename $f) done

If everything looks correct, try to rebuild the project via the rebuild button in the dashboard.

If everything has gone so wrong that you want to sit in a corner and cry... or restart from scratch

THIS SHOULD NOT BE DONE, but I document it in any case.

To undo a release, so that it can be replayed from the beginning you should (not that it is a somehow reverse order to what done in the normal procedure):

• uninstall from CVMFS... you shouldn't have gone that far...

• revert the software database (should not be needed)

• remove the old-style tarballs and html files

    rm -i $LHCBTAR/html/${MyProject^^}_${MyProject^^}_${MyVersion}*
    rm -i $LHCBTAR/${MyProject^^}/${MyProject^^}_${MyProject^^}_${MyVersion}*

• remove web page links

• uninstall the RPMs

  ♦ check that you know what you want to remove

    /afs/cern.ch/lhcb/software/lbinstall/lbinstall list ${MyProject^^}_${MyVersion}
For the Librarian

Releasing a new version of LCG externals

Until Gaudi v23r2

Until Gaudi v23r2, the dependencies to be included in LCGCMT were included in the Gaudi requirements files. The procedure to build the tarball was therefore the following:

For Gaudi, one more setup needs to be done: the generation of the tarball of its dependencies. This has to be done for each and every optimized >binary< (x86-64-slc6-gcc46-opt, x86-64-slc5-gcc46-opt, x86-64-slc5-gcc43-opt, i686-slc5-gcc43-opt)

```
cd $Gaudi_release_area
mkLCGCMTtar -n GAUDI_<version> -b <binary>
```

This script produces a log file in the local directory from where it has been run. Check it to see if everything was OK.

As from Gaudi v23r3...

The generation of the tarball of its dependencies. This has to be done for each and every optimized and debug >binary< (x86-64-slc6-gcc48-opt, x86-64-slc6-gcc48-dbg, x86-64-slc6-gcc49-opt, x86-64-slc6-gcc49-dbg).

The project LHCbExternals contains the list of dependencies to be included in LCGCMT. It is therefore necessary to:

1. Make sure that the dependency list is up to date
2. Tag a new version of LHCbExternals, with a version matching LCGCMT e.g. LCGCMT 63 -> LHCbExternals v63r0 LCGCMT 63a -> LHCbExternals v63r1 etc...

Release LHCbExternals to AFS:

```
cd /afs/cern.ch/lhcb/software/releases/
mkproject -p LHCbExternals -v vXrY -a ngc
mkproject -p LHCbExternals -v vXrY -a K
```

Now prepare the tar ball of LCGCMT, which has to be done differently depending on the version of LCGCMT:

* Up to LCGCMT 66:

```
cd /afs/cern.ch/lhcb/software/releases/
mkLCGCMTtar -n LHCBEVENTials_<version> -b <binary>
```

If everything has gone so wrong that you you want to sit in a corner and cry... or restart from scratch.
This script produces a log file in the local directory from where it has been run. Check it to see if everything was OK.

* As from LCGCMT 68:

  mkLCGCMTtarFromRPM LHcbExternals v69r0p1 $CMTCONFIG

As from LCG 86

We do not extract dependencies any more, the LHCBEVENTALS project is abandoned.

To Generate the install project tar files LCGCMT for x86_64-slc6-gcc49-opt, just run (LbScript v8r7 at least is needed)

  makeTarFromRPM LCGCMT v86r0 x86_64-slc6-gcc49-opt ./LHCBEXTERNALS_86.json

Where LHCBEVENTALS_86.json is a file of the form:

```json
{
    "heptools": {
        "version": 86,
        "packages": [
            "AIDA",
            "Boost",
            "eigen",
            "CLHEP",
            "COOL",
            "CORAL",
            "CppUnit",
            "Frontier_Client",
            "GSL",
            "HepMC",
            "HepPDT",
            "Python",
            "QTest",
            "Qt",
            "RELAX",
            "ROOT",
            "XercesC",
            "fastjet",
            "fftw",
            "graphviz",
            "libunwind",
            "neurobayes_expert",
            "oracle",
            "pyanalysis",
            "pygraphics",
            "pytools",
            "sqlite",
            "tcmalloc",
            "vdt",
            "xqilla",
            "xrootd",
            "tbb",
            "rangev3",
            "cppgsl",
            "ipython"
        ]
    }
}
```
As from LCG 87

We now just need to pre-install the build dependencies so that the nightly builds can continue working. The RPM needed at install time are pulled in as needed.

The metadata is contained in files such as:

LHCBEVENTALS_88.json:

```
{
    "heptools": {
        "version": 88,
        "packages": [
            "AIDA",
            "Boost",
            "eigen",
            "CLHEP",
            "COOL",
            "CORAL",
            "CppUnit",
            "Frontier_Client",
            "GSL",
            "HepMC",
            "HepPDT",
            "Python",
            "QTest",
            "Qt",
            "RELAX",
            "ROOT",
            "XercesC",
            "fastjet",
            "fftw",
            "graphviz",
            "libunwind",
            "neurobayes_expert",
            "oracle",
            "pyanalysis",
            "pygraphics",
            "pytools",
            "sqlite",
            "tcmalloc",
            "vdt",
            "xqilla",
            "xrootd",
            "tbb",
            "rangev3",
            "cppgsl",
            "ipython"
        ]
    }
}
```

The metadata files are kept and versioned in https://gitlab.cern.ch/lhcb-core/rpm-recipes (LHCBEVENTALS sub directory).

Follow the instructions in the README.md to get the list of rpms to install, and to install them on CVMFS and AFS.

Then update the software configuration DB, e.g.

```
lb-sdb-addplatform LCG 91 x86_64-slc6-gcc62-opt
lb-sdb-addplatform LCG 91 x86_64-slc6-gcc62-dbg
lb-sdb-addplatform LCG 91 x86_64-centos7-gcc62-dbg
```
Updating the Software Configuration DB for a new LCG or Gaudi

First import Gaudi without asking for a release:

```bash
lb-sdb-import --norelease Gaudi v28r3
```

If we are dealing with a new LCG, add the corresponding platforms:

```bash
lb-sdb-addplatform LCG 89 x86_64-slc6-gcc62-opt
lb-sdb-addplatform LCG 89 x86_64-slc6-gcc62-dbg
lb-sdb-addplatform LCG 89 x86_64-centos7-gcc62-dbg
lb-sdb-addplatform LCG 89 x86_64-centos7-gcc62-opt
lb-sdb-addplatform LCG 89 x86_64-centos7-gcc7-opt
lb-sdb-addplatform LCG 89 x86_64-centos7-gcc7-dbg
```

If Gaudi features more platforms than LCG, then specify the list of platforms to release:

```bash
lb-sdb-addplatform --release Gaudi v28r3 x86_64-slc6-gcc62-opt
```

Then schedule Gaudi for release:

```bash
lb-sdb-release Gaudi v28r3
```

Then check if everything is ok:

```bash
$ lb-sdb-query show Gaudi v28r3
Node 382601 Properties
-------------------------------
project   : GAUDI
version   : v28r3
-------------------------------
2683451:REQUIRES(O)     -> (ID:382602, project:LCG, version:89)
2683463:REQUESTED_PLATFORM(O) -> (ID:382306, platform:x86_64-centos7-gcc62-opt)
2683464:REQUESTED_PLATFORM(O) -> (ID:382607, platform:x86_64-centos7-gcc7-dbg)
2683466:REQUESTED_PLATFORM(O) -> (ID:382305, platform:x86_64-centos7-gcc62-opt)
2683467:REQUESTED_PLATFORM(O) -> (ID:382608, platform:x86_64-centos7-gcc7-opt)
2683468:REQUESTED_PLATFORM(O) -> (ID:382307, platform:x86_64-centos7-gcc62-opt)
2683469:REQUESTED_PLATFORM(O) -> (ID:382303, platform:x86_64-slc6-gcc62-dbg)
2683470:REQUESTED_PLATFORM(O) -> (ID:382302, platform:x86_64-slc6-gcc62-opt)
2683471:REQUESTED_PLATFORM(O) -> (ID:382609, platform:x86_64-centos7-gcc7-do0)
2683473:REQUESTED_PLATFORM(O) -> (ID:382304, platform:x86_64-slc6-gcc62-opt)
2683449:PROJECT(I)      <-- (ID:122379, project:GAUDI, sourceuri:gitlab-cern:gaudi/Gaudi)
2683474:RELEASEREQ(I)   <- (ID:122366, project:NONE, type:RELEASE, version:NONE)
```

Releasing a new version of LCG Grid

After LHCbDirac v8r0 (before then mkLCGCMTTar was necessary)

```bash
mkLCGCMTtarFromRPM  LHCbDirac vXrY $CMTCONFIG --nonatives
```

If some RPMs are missing from the middleware, it is possible to create them and to add them to the LHCb externals repository in $LHCBTAR/rpm/lcg The code necessary is in the LCGRPMP repo. It can be used to create the spec.
git clone https://gitlab.cern.ch/lhcb-core/LCGRPM.git
cd LCGRPM/package
./createExternalRPMSpec.py -s Grid -o ext.spec voms 2.0.12-3 x86_64-slc6-gcc48-opt
rpmbuild -bb ext.spec

After that step, the file should be copied to the RPM repo and the RPM repo should be reindexed:

```bash
cp /tmp/rpmbuild/RPMS/noarch/voms_2.0.12-3_x86_64_slc6_gcc48_opt-1.0.0-1.noarch.rpm $LHCBTAR/rpm/lcg
createrepo --workers=20 --update $LHCBTAR/rpm/lcg
```

N.B. /tmp/rpmbuild is a default of createExternalRPMSpec, and the -b option can be used to put the files to another directory e.g.

```bash
./createExternalRPMSpec.py -b /tmp/lben/toto -s Grid -o ext.spec voms 2.0.12-3 x86_64-slc6-gcc48-opt
```

Creates the RPM:

```bash
/tmp/lben/toto/rpmbuild/RPMS/noarch/voms_2.0.12-3_x86_64_slc6_gcc48_opt-1.0.0-1.noarch.rpm
```

Preparing the meta RPM with the software to be installed locally on the Online farm

First get a local version of LbNightlyTools:

```bash
mkdir -p /build/$USER
cd /build/$USER
git clone http://git.cern.ch/pub/LbNightlyTools
cd LbNightlyTools
./setup.sh
```

Now prepare the Meta RPM spec. There are two ways:

- By specifying explicitly the packages to be included:

  ```bash
  lbn-generate-metaspec -o ofm.spec OnlineFarmMeta 1.1.1 MOOREONLINE_v23r6_x86_64_slc6_gcc48-opt
  ```

  (You need to pass all necessary applications RPMs on the command lines, dependencies are dealt with automatically) And build it:

  ```bash
  rpmbuild -bb ofm.spec
  ```

- Using the ONLINE tag in the software configuration DB:

First add the project/version/combination to the soft configuration DB and check that the update worked:

```bash
lb-sdb-tag -p x86_64-slc6-gcc48-opt MooreOnline v23r7p15 ONLINE
lb-sdb-query listTag ONLINE
```

Removal can be done with

```bash
lb-sdb-tag -r -p x86_64-slc6-gcc48-opt MooreOnline v23r7pq5 ONLINE
lb-sdb-query listTag ONLINE
```

Then produce the RPM:

```bash
./lb-sdb-env.sh
lbn-generate-metaspec -o ofm.spec OnlineFarmMeta 1.1.1 -t ONLINE
```
(You need to pass all necessary applications RPM on the command lines, dependencies are dealt with automatically) Build the RPM:

```
rpmbuild -bb ofm.spec
```

Now copying it to the RPM repository:

```
```

Now install on plus:

```
cd /group/hltsw
./onlinelbpkr install OnlineFarmMeta
```

-- MarcoClemencic - 30 Jun 2014

-- MarcoCattaneo - 3 Jun 2017

-- MarcoClemencic - 2018-01-22