

Table of Contents

1 Persistency software release validation.....	1
1.1 Prerequisite: database account configuration.....	1
1.2 Execute the CORAL and COOL tests for a given platform.....	2
1.3 Analyse test results and commit them to SVN for reference.....	3

1 Persistency software release validation

This page describes how to run the CORAL and COOL tests to validate a production release installed on AFS or cvmfs. It was created to better document the latest release process since the move from CMT to cmake (see CORALCOOL-2896).

With respect to the previous release validation process using CMT (and dating back to 2013 approximately) described here, a few important differences apply:

- The validation procedure is performed (not on a systematic basis) by the CORAL/COOL team. For a brief period in the past it had been the responsibility of the SPI team, who now only take care of release builds and installations.
- The validation procedure does not require any write access to AFS (or cvmfs). The CORAL/COOL functional tests are executed in `/tmp` using the software installed on AFS or cvmfs.
- The validation procedure no longer includes the dedicated CoralServer test suite (see CORALCOOL-2905 and CORALCOOL-2906). It only includes the CORAL/COOL functional tests.

Note that the validation procedure can be used for any complete installation of the CORAL or COOL software. This includes nightly builds installed on AFS (eventually on cvmfs) and also private development builds, not only production releases, but it is worth noting the following subtle differences:

- As mentioned below, some changes were applied to trunk after LCG85, which is currently the latest installed production release. For this reason, some options in the procedure described below (in particular, the options to run CORAL tests with `andreaav` or `cdehort` credentials) are only fully functional on nightly builds and private trunk builds, or on production releases LCG86 and above.
- The validation procedure involves a full checkout of test logs in an area where new logs can be written. Production releases and nightly builds are installed on read-only areas of AFS and cvmfs, while development builds are installed on directories where developers have write access. For this reason the test suite scripts check out test logs under `/tmp` for production releases and nightly builds, and in the local project area for development builds.
- Production releases and nightly builds are built from code that cannot be modified until the next release or nightly build. The procedure described in the "Fix issues and rerun tests" section below only applies to private development builds (except for issues in authentication or dblookup that can be fixed also for releases and nightly builds).

1.1 Prerequisite: database account configuration

As described in `PersistencyTests`, access to Oracle and MySQL accounts and to properly configured Frontier and CORAL servers is needed to run the CORAL and COOL tests. On the client side, two files `authentication.xml` and `dblookup.xml` containing the relevant parameters must be available, in directories indicated by the `CORAL_AUTH_PATH` and `CORAL_DBLOOKUP_PATH`, respectively.

The nightly tests are automatically executed as Linux user `sftnight`, using a set of credentials stored in the `authentication.xml` and `dblookup.xml` files in directory `/afs/cern.ch/sw/lcg/app/pool/db`. Clashes between nightly tests simultaneously executed for different platforms in the same slot, or even for the same platform in different slots, are avoided by using different table names: this is implemented for both CORAL and COOL, using slightly different mechanisms. For CORAL C++ and Python tests, a unique table name is built based on the values of `CORAL_BINARY_TAG_HASH` (uniquely assi

To avoid clashes between automatically executed nightly tests and manually executed release validation tests, it is better to use two separate sets of accounts for these tests if possible. The database servers and accounts used for the nightly tests executed as Linux user `sftnight` are described in `PersistencyTestServers`.

In the following, `CORAL_AUTH_PATH` and `CORAL_DBLOOKUP_PATH` are set to point to `/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private`, where two minimal XML files used by members of the development team have been specially prepared. This directory is only readable by selected members of the development team. For simplicity, all database aliases are named after the Linux user `sftnight`, and for this reason it is necessary to set `COOL_QMTEST_USER=sftnight` before executing the COOL tests (see CORALCOOL-2888), but these are *not* the same accounts as those used by `sftnight` in the nightlies.

It should be noted that read access to `/afs/cern.ch/sw/lcg/app/pool/db` is no longer needed to execute the CORAL network glitch test (CORALCOOL-2929). Previously this path was hardcoded in the test, but it is now possible to run all tests with appropriate `authentication.xml` and `dblookup.xml` from any user-defined path. The change was implemented after LCG85, so read access to this directory is necessary to perform the validation of the LCG85 release.

1.2 Execute the CORAL and COOL tests for a given platform

The following is an example showing how to execute the CORAL and COOL tests on an SLC6 machine. Note that the CORAL and COOL tests for the same platform can be executed in parallel as they use different sets of tables, even when they use the same database accounts.

This procedure is only available and fully functional as of LCG_85, while older releases may be affected by some runtime configuration issues due to `lcgcmake`.

For CORAL:

```
bash
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
/cvmfs/sft.cern.ch/lcg/releases/LCG_85/CORAL/3_1_5/x86_64-slc6-gcc49-opt/qmtestRun.sh
```

For COOL:

```
bash
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
export COOL_QMTEST_USER=sftnight
/cvmfs/sft.cern.ch/lcg/releases/LCG_85/COOL/3_1_5/x86_64-slc6-gcc49-opt/qmtestRun.sh
```

Before launching the tests, the scripts above check out from SVN the most recent test logs into `/tmp/${USER}/LCG_85/CORAL/3_1_5/logs` and `/tmp/${USER}/LCG_85/COOL/3_1_5/logs`, respectively. The results of the tests will then be available in those directories, where they can be analyzed as described in the following subsections.

...

For CORAL trunk tests as `andreav`:

```
bash
export CORAL_QMTEST_USER=andreav
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private/db/$CORAL_QMTEST_U
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
export CORAL_NETWORKGLITCHTEST_USER=lcg_cool
/home/avalassi/CORAL/trunk/x86_64-slc6-gcc49-dbg/qmtestRun.sh
```

For COOL trunk tests as `andreav`:

PersistencyReleaseValidation < Persistency < TWiki

```
bash
export COOL_QMTEST_USER=andreav
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private/db/$COOL_QMTEST_US
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
/home/avalassi/COOL/trunk/x86_64-slc6-gcc49-dbg/qmtestRun.sh
```

For CORAL trunk tests as cdelort:

```
bash
export CORAL_QMTEST_USER=cdelort
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private/db/$CORAL_QMTEST_U
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
export CORAL_NETWORKGLITCHTEST_USER=lcg_pool_nightly
/home/avalassi/CORAL/trunk/x86_64-slc6-gcc49-dbg/qmtestRun.sh
```

For COOL trunk tests as cdelort:

```
bash
export COOL_QMTEST_USER=cdelort
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private/db/$COOL_QMTEST_US
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
/home/avalassi/COOL/trunk/x86_64-slc6-gcc49-dbg/qmtestRun.sh
```

...

For CORAL nightly tests as sftnight:

```
bash
export CORAL_QMTEST_USER=sftnight
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private/db/$CORAL_QMTEST_U
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
export CORAL_NETWORKGLITCHTEST_USER=lcg_coral_nightly
/afs/cern.ch/sw/lcg/app/nightlies/dev3/Sun/CORAL/3_1-patches/x86_64-slc6-gcc49-opt/qmtestRun.sh
```

For CORAL nightly tests as sftnight:

```
bash
export COOL_QMTEST_USER=sftnight
export CORAL_AUTH_PATH=/afs/cern.ch/sw/lcg/app/releases/CORAL/internal/private/db/$COOL_QMTEST_US
export CORAL_DBLOOKUP_PATH=$CORAL_AUTH_PATH
/afs/cern.ch/sw/lcg/app/nightlies/dev3/Sun/COOL/3_1-patches/x86_64-slc6-gcc49-opt/qmtestRun.sh
```

...

Test type
Releases (up to LCG85)
Releases (from LCG86)
Nightly builds
Private trunk builds

1.3 Analyse test results and commit them to SVN for reference

A short summary of test results is printed out by the scripts above when they terminate. The test suite should be considered successful when all individual tests either PASS or are UNTESTED, but there should be no tests that FAIL. As an example, the results of the CORAL 3.1.5 test suite in LCG85 are the following:

```
--- STATISTICS -----
      282      tests total
      10 (  4%) tests FAIL
```

PersistencyReleaseValidation < Persistency < TWiki

```
146 ( 52%) tests UNTESTED
126 ( 45%) tests PASS
```

After executing the test suites, the following script should be launched to produce dumps of the full test output in xml format (`x86_64-slc6-gcc49-opt.xml`) and detailed test summaries (`x86_64-slc6-gcc49-opt.summary`) from the raw qmtest dumps (`x86_64-slc6-gcc49-opt.qmr`). The xml and summary files can then be committed to SVN for reference, while the qmr file can be discarded.

```
cd /tmp/${USER}/LCG_85/CORAL/3_1_5/logs/qmtestCoral/
./summarizeAll.sh x86_64-slc6-gcc49-opt
svn commit x86_64-slc6-gcc49-opt.xml x86_64-slc6-gcc49-opt.summary
cd /tmp/${USER}/LCG_85/COOL/3_1_5/logs/qmtestCool/
./summarizeAll.sh x86_64-slc6-gcc49-opt
svn commit x86_64-slc6-gcc49-opt.xml x86_64-slc6-gcc49-opt.summary
```

If some of the tests failed, more details about the reasons for the failures can be analysed in the summary files.

-- AndreaValassi - 2016-07-28

This topic: [Persistency > PersistencyReleaseValidation](#)

Topic revision: [r11 - 2016-08-08 - AndreaValassi](#)



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? [Send feedback](#)