

Table of Contents

Persistency Framework project milestones.....	1
Persistency Framework project progress since 2010.....	1
2017 Q2 - 2017 Q3 (06 Oct 2017).....	1
2016 Q4 - 2017 Q1 (04 Apr 2017).....	1
2016 Q2 - 2016 Q3 (03 Oct 2016).....	2
2015 Q4 - 2016 Q1 (18 Mar 2016).....	3
2015 Q2 - 2015 Q3 (03 Oct 2015).....	3
2014 Q4 - 2015 Q1 (19 Mar 2015).....	4
2014 Q2 - 2014 Q3 (06 Oct 2014).....	4
2013 Q4 - 2014 Q1 (10 Apr 2014).....	5
2013 Q2-Q3 (27 Sep 2013).....	5
2013 Q1 (08 May 2013).....	6
2012 Q4 (23 Jan 2013).....	7
2012 Q3 (08 Nov 2012).....	8
2012 Q2 (26 Jul 2012).....	8
2012 Q1 (04 May 2012).....	9
2011 Q4 (27 Jan 2012).....	9
2011 Q3 (04 Nov 2011).....	10
2011 Q2 (20 Aug 2011).....	11
2011 Q1 (19 Apr 2011).....	12
2010 Q4 (10 Feb 2011).....	12
2010 Q3 (30 Sep 2010).....	12
2010 Q2 (12 Jul 2010).....	13

Persistency Framework project milestones

The development progress of the Persistency Framework project is regularly reported to the WLCG and AA management (normally every three months).

- Since 2010, PF progress is reported via a short textual contribution to the AA quarterly report (QR) to the WLCG management. These contributions are copied below for convenience.
- Between 2005 and 2010, PF progress was tracked using a milestone spreadsheet. The last such report was produced for Q2 2010 [↗](#).
- Progress earlier than 2005 is described in the original quarterly reports that are still available on the Applications Area Management page [↗](#).

The full WLCG quarterly reports since 2005 are all available in the WLCG Document Repository [↗](#).

Persistency Framework project progress since 2010

The following is a list of the PF contributions to the AA quarterly report to the WLCG management since 2010. These highlight the progress in software development, as well as in the areas of service operation and user support in a broader sense. More details about the software development progress is covered also in the PersistencyReleaseNotes.

2017 Q2 - 2017 Q3 (06 Oct 2017)

- One set of new release tags 3.1.9 for CORAL and COOL were released during Q3 2017. These include important fixes and improvements contributed by ATLAS for the CoralServer component, changes for Frontier suggested by CMS and a few configuration enhancements to support the upcoming AVX builds. The new tags have been used in the three release series LCG89, LCG90 and LCG91 against different sets of externals, including ROOT 6.10 with different patch versions and Boost 1.64. Support for gcc7 on both SLC6 and CentOS7 was added in these new releases.
- The migration of the CORAL and COOL repositories from SVN to gitlab has been completed.
- The port of CORAL and COOL to Python3 is in progress. As of LCG89, all new configurations of the LCG stack are released separately for both Python2 and Python3, but work is still needed on both the PyCoral and PyCool components to complete their port and integration in the Python3 infrastructure.

2016 Q4 - 2017 Q1 (04 Apr 2017)

See the full WLCG half-yearly report for 2016 Q4 - 2017 Q1 [↗](#).

- Three tags were released during Q4 2016 and Q1 2017: CORAL and COOL 3.1.6, 3.1.7 and 3.1.8. The number of platforms increased from 6 to 8 in the releases 3.1.6, and remained the same for the two other releases. The new platforms are gcc62 on SLC6 and Centos7. All releases use ROOT 6.08 with different patch versions, and Boost 1.62.
- Many of the fixes and improvements in the CORAL and COOL 3.1.6 releases were prepared in Q3 2016 and had already been described in the previous half-yearly WLCG Report. These include: a simpler and more standard implementation of the cmake infrastructure and lcgcmake integration; critical fixes in the CORAL server components; dropping support for legacy CORAL and COOL APIs; a comprehensive cleanup of test code.
- The 3.1.7 and 3.1.8 releases include many additional fixes and improvements, such as: further improvements in the cmake infrastructure (mainly for standalone builds outside lcgcmake); the full

removal of PyCintex from PyCool (using cppy only and dropping support for ROOT5); fixes in all plugins (including critical fixes for crashes in CoralAccess and FrontierAccess); improvements in the monitoring service (including fixes in its lifetime management); a further massive internal cleanup of the code (including the replacement of boost::shared_ptr by c++11 std::shared_ptr); improvements in tests.

- Some progress has also been done to port CORAL and COOL to Python3. The port of the internal CORAL and COOL cmake infrastructure has been achieved, while the port of PyCoral and PyCool is foreseen to be completed in Q3.

2016 Q2 - 2016 Q3 (03 Oct 2016)

See the full WLCG half-yearly report for 2016 Q2 - 2016 Q3 [↗](#).

- Two new release tags 3.1.4 and 3.1.5 of CORAL and COOL were prepared during Q2 2016. These contain many code and configuration changes to port CORAL and COOL to the latest MacOSX version 10.11, including work-arounds for several new issues specific to this O/S, such as System Integrity Protection runtime constraints for shared libraries and the use of a custom version of SQLite. Two pre-releases of the LCG85 stack requested by the SWAN project were built against 3.1.4, while the production LCG85 stack requested by ATLAS was built against 3.1.5. LCG85 is the first version of the stack that is also built and supported on Ubuntu. The port to gcc6.1 and clang 39 on Linux is also ongoing. The issues in path relocation for CORAL/COOL test and setup scripts observed in LCG84 have been fixed in the lcgmake version used for LCG85. This new release family includes major upgrades in many external packages, such as the move to Boost 1.61 and to MySQL 5.7.11 (which, however, continues to be built internally against the older Boost version 1.59). Many new external packages used by CORAL and COOL have also been added to the stack, as requested by the SWAN project and by the port to Ubuntu and MacOSX 10.11, even if on SLC6 and CentOS7 some of these new externals are already provided by the underlying O/S via the HEP_OSlibs meta package. In LCG85, crashes in PyCool may still be observed due to a bug in ROOT that will only be fixed in ROOT 6.08 in LCG86.
- Most of the work during Q3 2016 focused on the preparation of a new tag 3.1.6 of CORAL and COOL for the upcoming LCG86 release, including many fixes and improvements in both the code and configuration of the two packages. The build infrastructure has been significantly modified, to adopt more standard practices for cmake-based projects and simplify the integration of CORAL and COOL in the lcgmake machinery used for building all software packages in the LCG stack. A large number of implementation code fixes have been applied in both core packages and database plugins, including critical fixes for issues leading to server-side and client-side crashes in the CORAL server components. The code base has also been significantly simplified by dropping the switches that would still allow to build it against older versions of the CORAL and COOL APIs or without the c++11 compiler option. Finally, a further comprehensive cleanup of test code has been completed.
- The transfer of responsibility for the CoralServer components to ATLAS has been agreed and implemented. The software will continue to be hosted and built with the rest of CORAL, within the same repository and with the same test/release cycles and procedures, but the resolution of any issues specific to the CoralServer software will now be the responsibility of ATLAS. The dedicated CoralServer test suite for the ATLAS HLT use case has been moved to a different part of the repository under ATLAS responsibility and will no longer be included in CORAL releases. A new CORAL server instance for nightly and release tests has also been set up by ATLAS.
- The transfer of responsibility for the maintenance of the CORAL and COOL projects within CERN IT to the IT-DB group is now essentially completed. The main coding activities which were ongoing beyond ordinary maintenance (such as the implementation of a new lcgmake integration) have been completed in the current 3.1.6 release tag candidate. Hundreds of old JIRA tickets have been closed or

updated and a few more are being similiary handled. The developer-level twiki documentation has been significantly enhanced and is still being updated in coordination between the old and new project leaders.

2015 Q4 - 2016 Q1 (18 Mar 2016)

See the full WLCG half-yearly report for 2015 Q4 - 2016 Q1 [↗](#).

- Three new release tags 3.1.1, 3.1.2 and 3.1.3 of CORAL and COOL, all using ROOT6 and cmake, were prepared for ATLAS and LHCb during Q4 2015 and Q1 2016. Eight versions of the LCG stack, ranging from LCG81 to LCG84, were built based on these tags, against different sets of external packages and using different versions of lcgcmake. Several issues involving the build system, ROOT6/PyCool, SQLite and Frontier, details for which are given below, were addressed by these configurations: all of these issues have been fixed by the latest 3.1.3 releases in LCG84, with the exception of a few pending build system related issues that however have been fixed in the latest nightly builds and will be fixed in the next major LCG configuration. In addition: CORAL/COOL release 3.1.1 includes new PyCoral features requested by ATLAS online to prototype CPython-based bindings for COOL; 3.1.2 includes an important fix in CoralServerProxy and the port to c++14 and to the gcc52 compiler (with the old ABI); 3.1.3 includes the port to ARM and many related fixes for bugs in char handling that also affect x86 platforms.
- Several issues in the initial cmake-based build system that replaced CMT in LCG80 have been identified and fixed by patches in both CORAL/COOL and lcgcmake. As of LCG83, the compiler flags for CORAL/COOL are synchronised to those used in lcgcmake for the rest of the stack, ensuring that debug symbols are produced if needed and the appropriate c++11/c++1y/c++14 standard is used. Complex issues in path relocation with CORAL/COOL test and setup scripts affect all releases so far, including LCG84; these issues have only been completely solved in the current latest master of CORAL/COOL and lcgcmake, by completely disabling the lcgcmake post-install scripts developed for the genser project.
- The upgrade to ROOT 6.06.00 in LCG83 introduced a new critical issue, potentially leading to crashes in PyCool. This was fixed by the upgrade to ROOT 6.06.02 in LCG84.
- The upgrade to SQLite version 3090200 in LCG82 included important bug fixes and new features in SQLite, but led to regressions in CORAL, which were fixed in CORAL 3.1.2. The LCG stacks including LCG83 and above include both the fixes in SQLite and those in CORAL.
- The upgrade to frontier_client 2.8.18 in LCG84 fixes a large number of problems, including a critical issue in SSL when both frontier_client and xrootd libraries are used in the same executable.

2015 Q2 - 2015 Q3 (03 Oct 2015)

See the full WLCG half-yearly report for 2015 Q2 - 2015 Q3 [↗](#).

- Two new release tags 3.0.3 and 3.0.4 of CORAL and COOL, using ROOT6 and CMT, were prepared for ATLAS and LHCb during Q2 2015. Several releases were built based on these tags, against different sets of external packages. In particular COOL 3.0.4 was built both against successive patch releases of ROOT 6.02 (up to 6.02.12 in LCG78root6) and the first production release of ROOT 6.04 (6.04.02 in LCG79). The upgrade to ROOT 6.04 using ORCJIT (as opposed to ROOT 6.02 using JIT) finally solves a long-standing issue with Python crashes in PyCool when COOL C++ exceptions are thrown. The many workarounds that had been added to COOL for this issue in ROOT 6.02-based releases will be removed when all experiments have moved to ROOT 6.04 and support for ROOT 6.02 is dropped. Other notable changes in these software versions include the first releases on CERN CentOS7, the code and configuration port to clang60 on MacOSX, to gcc51 and to Ubuntu, fixes for

all residual defects found in Coverity scans and for other bugs, further progress in integrating CoralServer tests with more recent versions of the ATLAS HLT software, and further progress in replacing Boost by c++11 in the internal implementation of CORAL and COOL. All changes to the code bases in 3.0.4 were also backported to two final sets of tags for the CORAL 2.4 and 2.3 and the COOL 2.9 and 2.8 branches, based on ROOT5 and CMT.

- The main achievement for CORAL and COOL during Q3 2015 consisted in the move of the internal build and configuration system from CMT to cmake, and in its integration into the lcgcmake infrastructure for nightly and release builds. This was also accompanied by a major restructuring of the two SVN repositories, moving packages to allow direct checkouts into the desired directory structure, and freezing the obsolete development branches CORAL2 and COOL2 based on ROOT5 and CMT. The scripts to set up the runtime environment and run tests from cmake-based installations have also been improved and made fully relocatable, which considerably simplifies the procedure to reproduce and debug issues in the nightlies, particularly those coming from external packages such as ROOT. New candidate release tags 3.1.0 of CORAL and COOL, based for the first time on cmake and on the new SVN structure, have been prepared for an upcoming LCG80 configuration. The new tags also include improvements to PyCool and progress in the port to icc15 and clang35 on Linux.

2014 Q4 - 2015 Q1 (19 Mar 2015)

See the full WLCG half-yearly report for 2014 Q4 - 2015 Q1 [↗](#).

- Three new releases of the 3.0 branches of CORAL and COOL have been prepared for ATLAS, based on successive patch releases of ROOT 6.02. All changes to the code bases have also been backported to new releases of the CORAL 2.4 and COOL 2.9 branches, based on ROOT5. The main change has been the implementation of a PyCool patch to avoid crashes in PyROOT when C++ exceptions are thrown: this issue is expected to be fixed with the upcoming move of ROOT6 from JIT to ORCJIT, but a temporary workaround had to be added as this issue became a blocker for ATLAS and its priority was escalated. A second major improvement has been the integration of the CORAL and COOL test suites, still based on CMT, with the new nightly build infrastructure, now based on cmake. Other notable changes include the transfer of responsibility to ATLAS for the deployment of Oracle client-side configuration files, the integration of CoralServer tests with more recent versions of the ATLAS HLT software, bug fixes for the Frontier backend of CORAL and the porting to CERN CentOS7 and gcc49.
- The 3.0 branch of COOL is also being tested in the nightly build infrastructure against the ROOT6 master branch, which includes the move from JIT to ORCJIT and a major reimplementing of PyROOT. This change is expected to bring many improvements and simplifications in PyCool, but progress is presently prevented by a blocking bug in PyROOT.
- The migration of the CORAL server test infrastructure and CORAL/COOL test nodes from quattor to Puppet has been completed. The CORAL and COOL test suites have been adapted accordingly.

2014 Q2 - 2014 Q3 (06 Oct 2014)

See the full WLCG half-yearly report for 2014 Q2 - 2014 Q3 [↗](#).

- New major releases CORAL 3.0.0 and COOL 3.0.0 have been prepared in the LCG_69root6 configuration, based on the first production release 6.00.00 of ROOT6. They include backward incompatible changes in the APIs of both CORAL and COOL, such as the complete removal of Boost and its replacement by C++11 classes. With respect to ROOT5-based releases, the PyCool package (for interactive Python usage of COOL) is now loading C++ headers at runtime through JIT, rather than at build time through genreflex; one major issue with C++ exception handling in PyCool is still pending, but this will only be fixed when ROOT6 moves from JIT to MCJIT. Tags CORAL 3.0.0a

and COOL 3.0.0a have also been prepared for the upcoming LCG_70root6 release that will be built against ROOT 6.02.00; the main difference with respect to LCG_69root6 tags lies in the way PyCool loads C++ headers, which has changed since ROOT 6.00.00.

- To allow the experiments to perform detailed comparisons between ROOT5 and ROOT6, CORAL and COOL branches compatible with ROOT5 are still being maintained, without C++11 extensions in their APIs. In particular, release tags CORAL 2.4.3 and COOL 2.9.3 have been prepared for the upcoming LCG_70root5 configuration; their code bases are equivalent to those of CORAL 3.0.0a and COOL 3.0.0a, except for the APIs that are still the same as in CORAL 2.4.2 and COOL 2.9.2.
- The migration from the two savannah trackers for COOL and CORAL/POOL to a new single JIRA tracker has been completed. Over 2700 issues have been migrated.

2013 Q4 - 2014 Q1 (10 Apr 2014)

See the full WLCG half-yearly report for 2013 Q4 - 2014 Q1 [↗](#).

- New major releases of CORAL and COOL have been prepared for ATLAS in the LCGCMT_67 configuration to provide new features in both packages, such as support for a new relational schema with "vector payload" in COOL and protocol changes and threading improvements in the CORAL server. These and other enhancements required backward-incompatible changes in both APIs, which were not possible during the LHC data taking and have been postponed to the current LS1 phase. This is the first release that is only supported with c++11 build options (using gcc47 and gcc48 on SLC6), although c++11 extensions are still disabled in CORAL and COOL as they would not be compatible with ROOT5. Two patch versions with urgent bug fixes in COOL query performance (LCGCMT_67a) and in the frontier client (LCGCMT_67b) have also been prepared.
- The port of PyCool to the latest ROOT6 beta3 version has been completed. Several issues in successive ROOT6 candidates have been identified and solved during this process, thanks to the good coverage of the COOL test suite. A release candidate LCG_68_root6 based on ROOT6 beta3 and a reference release LCG_68 based on ROOT5 have been produced for LHCb, using almost identical code bases for CORAL and COOL, except for the replacement of Boost by c++11 classes in the COOL API used with ROOT6. One major issue with c++ exception handling in PyCool is still pending, but this will only be fixed when ROOT6 moves to a more recent JIT version.
- Progress is being made in the port of the CORAL and COOL build system from CMT to cmake, in the port of the ticketing system from savannah to JIRA, and in the port of the CORAL server test infrastructure from quattor to Puppet.

2013 Q2-Q3 (27 Sep 2013)

See the full WLCG half-yearly report for 2013 Q2-Q3 [↗](#).

- The CORAL and COOL code repositories have been moved from CVS to SVN. The old CVS repositories have been closed and SVN is now used for committing all new developments. The full code history was migrated to preserve read-only access to historical data for later reference, as the CVS servers will soon be shut down. For the same reason, the POOL repository was also migrated, even if new developments have already been moved to the ATLAS SVN repository.
- New releases of CORAL and COOL have been prepared in Q2-Q3 2013 for ATLAS and LHCb (LCGCMT_65a and LCGCMT_66). LCG_65a is the first release based on the SVN repositories and includes only minor configuration fixes in the two packages. LCG66 is the first production release supported for the gcc4.8 compiler on SLC6 and includes important enhancements in both CORAL and COOL for adding support and optimizing query performance on Oracle 12c servers. It also

includes user-requested fixes, as well as the upgrade to Boost 1.53 and the port of the code base to the latest clang33 compiler.

- COOL query performance has been fully validated on Oracle 12c servers. The main result of these tests is that the "adaptive optimization" new feature of Oracle 12c is better kept disabled in COOL in order to provide stable and predictable performance. This led to a few changes in both the CORAL and COOL code bases. All 9 major read-only use cases of COOL have been tested and it was confirmed that good execution plans and scalable performance can now be achieved.
- Progress is being made on the port of PyCool to the upcoming ROOT6 major release, which will involve several changes in the pythonization of C++ code. In particular, direct C++ dependencies on Reflex have been successfully removed, while the migration from the PyCintex to the cppy python package is ongoing.
- Failures in the CORAL and COOL nightlies (ORA-12638) observed after the move of the test infrastructure to new (improperly configured) virtual machines have led to the discovery of previously unknown authentication issues in the Oracle client software. These are now being followed up, both for the current 11g client and for the new 12c client that is also being tested in view of its eventual deployment in the software stack.
- Progress is being made towards new major CORAL and COOL releases with API extensions and new functionalities, most notably the COOL vector payload schema. In particular, work is ongoing to validate COOL query performance for this new use case.

2013 Q1 (08 May 2013)

See the full WLCG quarterly report for 2013 Q1 [↗](#).

- New releases of CORAL and COOL have been prepared in Q1 2013 for ATLAS and LHCb (LCGCMT_64d, LCGCMT_61g, LCGCMT_65rc1 and LCGCMT_65). These releases introduce important bug fixes and enhancements in both CORAL and COOL, including fixes for memory leaks identified through valgrind and several improvements in the CORAL_SERVER package (also relevant to the eventual release of a new network protocol for client/server communication). The build and test configuration has also been adapted to changes in the LCGCMT policies and in the nightly test infrastructure. LCG65 is the first production release supported for the gcc4.7 compiler on both SLC5 and SLC6, using the c++11 standard, and it also includes the upgrade to Python 2.7, Boost 1.50 and the EMI2 Grid packages. The code bases have also been ported to the latest gcc4.8, icc13 and clang32 compilers.
- More progress has been made in the use of several profiling tools (valgrind, igprof, gperftools) to detect memory and time performance issues in CORAL and COOL. In particular, a dedicated suppression file for valgrind has been prepared and this tool has been integrated in the CORAL and COOL test suites to automatically obtain memory analysis reports. Thanks to the use of these tools, several memory leaks and other issues have been identified, many of which have already been addressed in the most recent software releases.
- The 11.2.0.3.0 Oracle instant client for MacOSX has been installed on the AFS external software area of the LCG AA, in collaboration with IT-DB. The new client is used by the latest LCGCMT_65 release and will be used on all future releases of the CORAL software. The availability of this client finally completes the move of all platforms to the Oracle 11g client (previously the 10g instant client was still used on MacOSX).
- Support has been provided to ATLAS to help them in their ongoing review of COOL usage during LS1. An API change in COOL will be required in one of the next releases to address some of the

issues raised in that context. An initial R&D on CORAL server monitoring using Hadoop has also been conducted.

- Working jointly with IT-GT, new versions of the HEP_OSlibs 'meta-rpm' for SLC6 and of the corresponding quattor profile have been prepared. This meta-package is simply a list of all packages required by the four LHC experiments on their Linux boxes in addition to the minimal SLC6 installation, to be able to run their reconstruction and analysis software, including CORAL and COOL. As agreed within the WLCG SL6 deployment task force, the meta-rpm has been made available on a dedicated 'wlcg' software repository on linuxsoft.cern.ch and a twiki page has been prepared to document the instructions for its installation.

2012 Q4 (23 Jan 2013)

See the full WLCG quarterly report for 2012 Q4 [↗](#).

- New releases of CORAL and COOL have been prepared in Q4 2012 for ATLAS (LCGCMT_61f) and LHCb (LCGCMT_64b and LCGCMT_64c), including important fixes and enhancements in both packages, as well as the upgrade to the Oracle 11.2.0.3.0 client, which provides the fix for a security vulnerability (CVE-2012-3137) in the Oracle logon protocol. The three new releases are all based on the same code bases for CORAL and COOL, using different external packages in the LCGCMT configuration. A major enhancement in both packages is the implementation of support for Oracle authentication using Kerberos (more details below). An improved handling of database connection glitches in the core CORAL software and in the CoralServerProxy component have been included to address issues observed in the ATLAS HLT system (more details below). Both packages also include several fixes for issues reported by the Coverity static code analyzer, as well as fixes for a few memory leaks that were identified thanks to the integration of valgrind in the test suites. Finally, the code bases have been ported to c++11 in gcc4.7 and to a more recent version of Boost.
- A new functionality for connecting to Oracle database servers using Kerberos authentication has been added to the latest CORAL and COOL releases. Two options are supported, authentication of external users (Kerberos principal is a new Oracle user/schema name) and external authentication of proxy users (Kerberos credentials are used to connect as an existing Oracle user/schema). These mechanisms (especially proxy authentication) could represent useful alternatives to user/password authentication at CERN, both for individual users and for those cases where shared passwords are used by several members of an LHC experiment team and are not encapsulated within a Frontier or CORAL server. Tests prepared with the help of IT-DB and IT-CF confirmed that it is now possible to connect to properly configured databases using the standard Kerberos ticket from the CERN KDC (i.e. the one also used for AFS).
- Support was provided to ATLAS in the analysis of a crash of the CORAL server during data taking with beam in November 2012, which was fixed by a clean restart of the whole HLT system. As in previous occurrences of this issue, the root cause of the problem was the loss of the connection to the ATONR Oracle server due to a 'network/database glitch' (which we are still unable to explain). A major patch for these issues is available in CORAL since Q2 2012, but is not yet used in ATLAS. Due to the frequency of these issues, this patch is now being considered for deployment in production in the ATLAS online software. Extensive tests of this patch by the ATLAS and CORAL experts led to a few additional enhancements in the core CORAL packages and the CoralServerProxy component in the LCGCMT_61f release. In this context, some memory leaks have also been identified and fixed, thanks to the work performed to integrate the valgrind memory checker with the CORAL and COOL test suites and to prepare valgrind suppression files for Oracle and the other external dependencies.
- The Oracle 12c beta client software has been tested for issues that had been observed in the use of the older 11g client with CORAL and COOL. The main focus of the tests was the redefinition in Oracle of GSSAPI and Kerberos symbols from the O/S libraries. It was found that the situation has

somewhat improved with respect to 11g, but some issues are still pending and are being followed up in an Oracle Service Request with the help of IT-DB.

- Working jointly with IT-GT, a first prototype of the HEPOS_libs 'meta-rpm' for SLC6 and of the corresponding quattor profile has been prepared. This meta-package is simply a list of all packages required by the four LHC experiments on their Linux boxes in addition to the minimal SLC6 installation, to be able to run their reconstruction and analysis software, including CORAL and COOL (as was done in the past for SLC5). The meta-rpm has been installed on a dedicated VObox where test accounts have been opened for computing experts of the four experiments and of the LCG AA projects.

2012 Q3 (08 Nov 2012)

See the full WLCG quarterly report for 2012 Q3 [↗](#).

- No new releases of the Persistency Framework projects have been built in Q3 2012. The last software versions built in Q2 2012 (for LCGCMT_63) have been re-released using new configurations (LCGCMT_64 and LCGCMT_64a) providing a few external software upgrades (e.g. frontier_client) for ATLAS and LHCb. The port of the CORAL and COOL code base to gcc4.7 with c++11 support has been completed, except for the removal of language constructs that are deprecated in c++11. Work is also ongoing on the preparation of new major releases with API changes in both packages, to be picked up by the experiments during the LS1 shutdown in 2013.
- A new Oracle client (11.2.0.3.0) has been installed on AFS in view of its use by the LHC experiments in the upcoming releases of CORAL. The main motivation for the upgrade is a critical fix for the security vulnerability CVE-2012-3137 in the Oracle logon protocol (a temporary server-side workaround has been deployed at CERN, but all applications will need to upgrade to this version of the Oracle client by April 2013). It has also been verified that the new software includes fixes for SELinux and AMD/multicore related bugs previously affecting the unpatched 11.2.0.1.0 Oracle client. It was found, however, that the 11.2.0.3.0 client is still affected by the redefinition of Kerberos and GSSAPI symbols, a bug which was observed in the past to cause issues due to clashes with the same symbols from the system libraries. A Service Request is open with Oracle about this problem and continues to be followed up.
- The CORAL server and MySQL server for the CORAL and COOL nightly tests, previously hosted on old hardware that had to be retired, have been moved to a new fully quattorized SLC5 virtual machine in the CERN computer centre.

2012 Q2 (26 Jul 2012)

See the full WLCG quarterly report for 2012 Q2 [↗](#).

- New releases of CORAL and COOL have been prepared in Q2 2012 for LHCb (LCGCMT_63), mainly motivated by the upgrade to ROOT 5.34. The CORAL release includes major improvements in the handling of connection instabilities (CORAL is now able to reconnect transparently if network glitches do not break a transaction context), as well as important fixes in the cleanup of stale OCI sessions (avoiding crashes reported in a few uncommon situations). This is also the first release on SLC6 and the first release where support for the LFC replica service component of CORAL has been dropped. Finally, the code base of CORAL and COOL has been ported to gcc4.7.
- Several patches have been applied to the CORAL and COOL code bases in order to address the issues (more than 700 for the two projects together) reported by the Coverity static code analyzer. Most issues have been fixed by CORAL or COOL patches that will be included in one of the next releases, others have been dismissed as due to bugs in some of the external dependencies (e.g. ROOT or

Boost). There are now no pending issues reported by Coverity left in CORAL or COOL.

- In collaboration with IT-DB, the possible use of Kerberos authentication for Oracle databases has been investigated. A test setup was successfully prepared to connect to a test database using the standard Kerberos ticket from the CERN KDC (i.e. the one also used for AFS). The feasibility of integrating Kerberos authentication into CORAL and COOL will now be investigated.
- Support was provided to LHCb about the problems they experienced when trying to connect to Gridka databases using CORAL. The problem is now understood as being due to the Oracle character set used at Gridka (WE8ISO8859P15), which is different from the one used at CERN and expected by CORAL (WE8ISO8859P1). Two possible solutions have been suggested by the CORAL team: the "P15" character set can be kept on the server only if a different version of the Oracle client (leading to larger client-side memory footprints) is used in CORAL; alternatively, the server-side character set should be changed back to "P1". The issue is now being followed up within LHCb.
- The COOL nightly tests have been failing repeatedly during Q2 2012 due to ORA-04031 errors (failure to allocate shared memory) on the test2 Oracle database. The issue was eventually fixed by the DBAs in IT-DB by changing the server-side memory configuration and restarting the database. This issue was difficult to debug and it is not excluded that it may show up again in the future, as the problem is due to memory fragmentation that builds up over time and has been observed to occur only after a few weeks since instance reboot. As a side effect of this incident, a possible performance bug in Oracle (not using bind variables in some internal queries) was also discovered and has been reported to Oracle as a Service Request by the CORAL team.

2012 Q1 (04 May 2012)

See the full WLCG quarterly report for 2012 Q1 [↗](#).

- Four new releases of the Persistency Framework projects have been prepared in Q1 2012 for ATLAS (LCG_61d, LCG_61e), CMS (CORAL 2.3.21) and LHCb (LCG_62b). All these releases include code base and configuration changes to complete the port of CORAL and COOL to Oracle 11g servers (working around an Oracle feature causing frequent ORA-01466 errors in the nightly tests). The LCG_61d and LCG_61e releases, motivated by urgent bug fixes in POOL for ATLAS, also include a few external package upgrades (uuid, frontier_client); LCG_61e is actually an urgent rebuild of LCG_61d, after downgrading back the version of ROOT to 5.30.05 because 5.30.06 breaks binary compatibility due to some API changes. The CORAL 2.3.21 release includes a possible fix for some ORA-25408 errors observed by CMS during database and network instabilities. The LCG_62b release, motivated by an urgent bug fix in ROOT, includes several additional fixes and improvements in CORAL and COOL, completing the port of the two packages to the clang30 compiler on SLC6.
- Major progress has been made also in the improvement of the reconnection mechanism in CORAL to react to network and database glitches. The old reconnection mechanism (valid for all backends but affected by several bugs) has been completely replaced by a new reconnection mechanism, which is only valid for the Oracle plugin but is much more stable and better tested. This work will be included in the next CORAL release 2.3.23, under preparation.

2011 Q4 (27 Jan 2012)

See the full WLCG quarterly report for 2011 Q4 [↗](#).

- The validation of COOL performance on Oracle 11g servers has been completed, confirming that COOL queries exhibit good performance and scalability on 11.2.0.3 for all COOL use cases. The poor performance previously observed on 11.2.0.2 servers is finally confirmed to be due to an Oracle bug (10405897), absent in 11.2.0.1 and fixed in 11.2.0.3. Help from IT-DB in the analysis of these issues

is kindly acknowledged. A summary of these tests, with links to detailed performance reports, is available at <https://twiki.cern.ch/twiki/bin/view/Persistency/CoolPerformanceTests>.

- New releases of all PF projects have been prepared for ATLAS and LHCb in Q4 2011 for the five new configurations LCG_61b, LCG_60e, LCG_62, LCG_61c and LCG_62a. Changes to the PF code bases (such as important fixes in CORAL and COOL for the upgrade to Oracle 11g servers), were included in several of these releases, while others involved at most a rebuild against new versions of ROOT and other external software packages. LCG_61b includes the first releases of CORAL and COOL fully installed on AFS and validated by the SPI team, using the detailed documentation previously prepared to this end by the PF team. LCG_62 is the first release that does not include POOL (as discussed below); it also includes the first production build with the gcc4.6 compiler on SLC5, a preliminary step to the release of the software using this compiler on SLC6.
- The issue of POOL support has been clarified with LHCb and ATLAS, in the context of the WLCG Data Management TEG. LHCb has already stopped using POOL, while ATLAS will continue to use it and need support for as long as the 2012 production version of the ATLAS software, based on the LCG61 series, is actively used. ATLAS will no longer need support for POOL for their releases based on LCG62, where a custom software package derived from POOL will be built and maintained by ATLAS as part of their internal software. The first such release already exists and will be used as an ATLAS development release in 2012; this will eventually become the production version of the ATLAS software, by end 2012 or beginning 2013.
- The PF team also contributed to the WLCG Database TEG in Q4 2011, particularly for what concerns the review of the ATLAS, CMS and LHCb conditions databases and the preparation of the relevant report. The issue of CORAL and COOL support is also being discussed in that context.
- Other activities in this quarter included user support to CMS, about their observation of ORA-25408 errors in the online Oracle cluster, and several discussions with the ATLAS TDAQ team to discuss the requirements and possible implementation of a monitoring infrastructure for the CoralServerProxy instances deployed in the ATLAS HLT system.

2011 Q3 (04 Nov 2011)

See the full WLCG quarterly report for 2011 Q3 [↗](#).

- New releases of the PF projects have been prepared for ATLAS in Q3 2011 for the two new configurations LCG_60d and LCG_61a. The new releases were motivated by the upgrades to the latest version of ROOT in the relevant branch (5.28.00g and 5.30.02, respectively). The LCG_60d release also includes the upgrade to a newer frontier_client with important performance optimizations and several minor improvements in CORAL and POOL. The LCG_61a release also includes important performance optimizations in CORAL (reducing the number of Oracle data dictionary queries), improvements in the internal handling of transactions in COOL (to prepare to expose transaction control in the user API), as well as minor patches to port all packages to the gcc4.6 compiler and SLC6 (except for the CORAL and POOL packages that depend on LFC, which have been disabled as the LFC client software is not yet available on SLC6). A third configuration LCG_59d for ATLAS, using ROOT 5.26.00g, did not trigger the rebuild of any PF package instead. Several long-standing issues in the automatic nightly builds and tests of the software have also been fixed with the help of the SPI team.
- The CORAL server and MySQL server for the CORAL and COOL nightly tests have been moved from an office desktop to a fully quattorized node in the CERN computer centre.
- Support is being provided to CMS by the CORAL team in IT-ES, in collaboration with IT-DB, to follow up the incidents that affected the CMS online Oracle databases in Q3 2011. On the client side

PersistencyMilestones < Persistency < TWiki

these incidents triggered ORA-25408 errors, signalling that update transactions were lost in an unrecoverable way. The focus has been on trying to reproduce the errors, using custom CORAL tests combined with server-side actions. The present understanding is that these problems are caused by server-side or network issues external to CORAL, but possible enhancements to CORAL to better handle these issues are still being investigated.

- Support is being provided to ATLAS by the CORAL team, in collaboration with other colleagues in IT-ES and IT-DB, to understand and work around the problems they are observing in accessing their Oracle conditions data from T0 jobs. The causes for the spikes of high load observed on the database servers, which result in the failures of many T0 jobs, are not yet understood and are being investigated. As a workaround, the CORAL team is helping ATLAS in evaluating the use of the Frontier/Squid or CORAL server/proxy caching technologies to avoid direct Oracle connections in T0 jobs. Initial tests with Frontier performed by ATLAS also led to the observation of a discrepancy between physics results when retrieving conditions via Oracle or via Frontier: this is now understood as a data caching bug in the ATLAS muon software, which is being addressed.
- The validation of COOL query performance on Oracle 11g servers has started, with help from the DBAs in both ATLAS and IT-DB. Tests have shown that a different execution plan with non-scalable performance (query times increase as IOVs are retrieved from larger tables) is obtained on 11g servers out-of-the-box, for the same exact SQL queries as on 10g servers. While many more tests are needed, it presently seems that adequate performance and scalability on 11g servers can only be obtained by forcing the use of the 10g query optimizer. If confirmed, the solution will be deployed in a new COOL release, while the issue will be reported to Oracle Support for further investigations.

2011 Q2 (20 Aug 2011)

See the full WLCG quarterly report for 2011 Q2 [↗](#).

- New releases of the PF projects have been prepared in Q2 2011 for the two new configurations LCG_60c for ATLAS and LCG_61 for LHCb. Both releases, motivated by the upgrades to new versions of ROOT (5.28.00e and 5.30.00, respectively), include major changes in CORAL and several fixes and enhancements also in COOL and POOL. These releases also include a new Oracle client configuration (11.2.0.1.0p3), to work around the redefinition in the Oracle client of some kerberos symbols, conflicting with those in the system libraries.
- The new CORAL 2.3.16 code base, from which both LCG_60c and LCG_61 have been built, includes a major internal reimplementations of the OracleAccess plugin. This patch fixes several crashes that may take place when manipulating queries created on sessions that have become invalid, in both single-threaded and multi-threaded use cases. A subset of these issues has also been fixed in the SQLiteAccess plugin, whereas the reimplementations of the other plugins is pending.
- Using these two releases as an example, the post-install validation of the CORAL, COOL and POOL builds by the PF team was also documented in detail to allow its partial automatization by the SPI team, to reduce even further the time to deliver new releases in the future.
- In collaboration with the ROOT team and several teams in IT, the PF team was also active during Q2 2011 in following up a service incident affecting the Kerberos KDC at CERN, that had initially been reported as possibly caused by the POOL software. The main cause of the problem was eventually identified as a bug in the xrootd client; a fix for this issue, already included in ROOT 5.28, was picked up by ATLAS in a backport to ROOT 5.26.00f (which was kept binary compatible and did not trigger a full rebuild of the CORAL/COOL/POOL stack).
- Finally, the future of POOL was discussed with ATLAS and LHCb during Q2 2011, motivated by the recent decision by LHCb to drop POOL in favor of direct ROOT access, which will leave ATLAS as

the only user of POOL. It is likely that the POOL code and the responsibility for its support will be taken over by ATLAS, on the timescale of the LHC shutdown in 2013 or possibly earlier. The long-term support model for CORAL and COOL will also need to be separately discussed with IT and the experiments (CORAL is used by ATLAS, CMS and LHCb; COOL by ATLAS and LHCb).

2011 Q1 (19 Apr 2011)

See the full WLCG quarterly report for 2011 Q1 [↗](#).

- New releases of the PF projects have been prepared in Q1 2011 for the two new configurations LCG_60a and LCG_60b for ATLAS. Both releases were motivated by the upgrades to new patches of ROOT 5.28 and by fixes in other external packages (frontier_client and Qt, respectively). LCG_60a also includes fixes in POOL (to port it to the latest ROOT) and COOL (for multi-threaded applications), while LCG_60b includes several fixes in CORAL and a few minor additional enhancements in POOL.
- A major internal reimplementaion of CORAL plugins is also underway to fix several crashes reported by the users and/or observed in internal tests. These crashes generally take place when manipulating queries created on sessions that have become invalid, either because they have been closed by the users or because they have been interrupted by network glitches and automatically restarted by CORAL in an incorrect way. A first version of the new implementation is essentially ready for the Oracle plugin to fix both single-threaded and multi-threaded issues and is now being ported to the other plugins. These patches will be included (in one or more steps) in the next CORAL releases.

2010 Q4 (10 Feb 2011)

See the full WLCG quarterly report for 2010 Q4 [↗](#).

- New releases of the PF projects have been prepared in Q4 2010 for the two new configurations LCG_59b (for ATLAS, based on ROOT 5.26) and LCG_60 (for LHCb, based on ROOT 5.28), using the same code base for both (COOL 2.8.8, CORAL 2.3.14 and POOL 2.9.11). The new releases include several bug fixes and enhancements in all three packages, mainly in CORAL (including a major restructuring of the test infrastructure to extend its coverage, as well as the implementation of a workaround for a bug causing endless connection retrial loops after a network glitch), but also in POOL (support for the latest I/O optimizations in ROOT and enhancements in the collections packages) and COOL (bug fixes for NaN handling). As in the past, the code is supported on Linux SLC5 with gcc4.3, MacOSX and Windows using vc9, while support for SLC4 and Windows vc7 has been dropped; production support for the Intel icc compiler has also been added for the first time on request from LHCb, and CORAL has been ported to the c++0x standard in gcc4.5 on request from CMS.

2010 Q3 (30 Sep 2010)

See the full WLCG quarterly report for 2010 Q3 [↗](#).

- Three new releases of the Persistency Framework projects have been prepared in Q3 2010 for LHCb (LCG_58e), ATLAS (LCG_59a) and CMS (CORAL 2.3.12). One important issue in Oracle database services was also analyzed and solved with the help of the PF team.
- The motivation for LCG_58e (July 2010) was to urgently fix the problems reported by LHCb in the CORAL LFCReplicaService component. The cause of the problem is a bug in Globus, which uses its own version of the gssapi library that is incompatible to the MIT version provided by the system. The problem, triggered by the upgrade in LCG_58d to Xerces 3.1, was fixed by the workaround of

removing the gssapi dependency from Xerces in LCG_58e. The real fix will consist in the upgrade to Globus 4.2 using versioned symbol for gssapi. The fix has already been validated using prototype builds of Globus 4.0.8 and of the middleware provided by VDT and IT-GT, but will not be deployed in production before the EMI release in April 2011. The Xerces patch was also applied to the previously released LCG_59 for ATLAS.

- While investigating the gssapi issue in Globus, it was found that the Oracle client libraries also reimplement their third different version of gssapi symbols, that conflict with both system and globus gssapi. A Service Request has been opened with Oracle to follow up this issue, proposing to use versioned gssapi symbols as in the latest Globus patch. The problem has not been reported to cause any issue so far, but it was tested that this could lead to problems in client applications depending on the order in which libraries are loaded at runtime.
- The LCG_59a release for ATLAS (July 2010) includes several enhancements in CORAL (functional fixes for Frontier, fixes of memory leaks for Oracle), COOL (support for a new relational schema with 'vector payload') and POOL (fixes in collections and in the test infrastructure). It also involves the upgrade to the voms 1.9.17 client software, with new functionalities required to support certificate authentication in the CORAL server software.
- The CORAL 2.3.12 tag prepared for CMS includes the fix for a crash observed with optimized gcc 4.3 builds, which is now understood to be caused by a bug in the OracleAccess plugin. The crash had already been observed last year, but had initially been attributed to a bug in gcc optimization and had been solved by the workaround of disabling optimization on one C++ file. The move to 2.3.12 is a major upgrade for CMS, which was previously using a one-year old version of CORAL. Discussions are ongoing with CMS to integrate CORAL in their nightlies to speed up the adoption of new CORAL versions in the future.
- Server-side process crashes (ORA-07445) triggered by COOL applications were observed on the ATLAS and LHCb databases, after the Oracle security updates were applied in June. Investigating the issue in collaboration with the IT-DB and ATLAS DBAs, the Persistency team prepared a COOL-based stress test suite that was successful in reproducing the issue on a test database. The tool was then used to validate the last patch proposed by Oracle to fix the issue, which could then be applied to the production services.

2010 Q2 (12 Jul 2010)

This is the same contribution as in the PF spreadsheet for 2010 Q2 [↗](#). See also the full WLCG quarterly report for 2010 Q2 [↗](#).

- Two new releases of all PF projects have been prepared for ATLAS in Q2 2010. The main motivation for LCG_56g (May 2010) was the CORAL upgrade to version 2.7.14 of the frontier_client library, to fix a wrong libexpat.so dependency which had triggered the failure of some ATLAS jobs accessing conditions data on the Grid. The LCG_59 release (July 2010) was motivated by major upgrades in many external dependencies (including ROOT, python and oracle), functionality enhancements in POOL collections and bug fixes in CORAL. Functionality enhancements are being prepared for ATLAS offline users also in CORAL and COOL, but their release has been postponed because they involve API extensions which would break binary compatibility with the ATLAS online software in the HLT system. A new LCG_58d is also being prepared (July 2010) for LHCb, using for all PF projects the same code base used in the ATLAS LCG_59 release. Following the upgrade of ATLAS to the same ROOT 5.26 code base as LHCb, the only difference between these two configurations is that ATLAS and LHCb will use python 2.6 and 2.5, respectively. In particular, both LCG_59 and LCG_58d use a new '11.2.0.1.0p2' version of the oracle client software, which completes the SELinux fixes for the 32 and 64 bit versions of the OCI and OCCI libraries and also contains the fix for the Oracle 11g bug on AMD multicore hardware, which had triggered the temporary downgrade of

ATLAS to the 10g client in Q1 2010.

- The use of Frontier for conditions data access in ATLAS (mainly for analysis jobs at T2 sites) is steadily increasing. While the integration of Frontier into the ATLAS client software in Q4 2009 had been very smooth as this backend was already fully supported both in CORAL (for CMS) and in COOL (just in case one experiment should need it), Frontier had never been tested against the ATLAS production use cases and a few ATLAS-specific optimizations in the Persistency Framework software have been requested. The latest such improvements to the FrontierAccess plugin are included in the LCG_59 release of CORAL.

-- AndreaValassi - 17-Sep-2010

This topic: Persistency > PersistencyMilestones

Topic revision: r40 - 2017-10-11 - AndreaValassi



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback