

PyCool progress this week (since Wed 20th)

- Worked in two directions in parallel
 - New CORAL240 and COOL290 releases for ATLAS
 - More changes in PyCool and debugging for ROOT6 beta
- A lot of progress on both fronts
 - Thanks to the help from many people!
 - But mixing LCG67 release and ROOT6 caused overheads...

Updates on issues pending last week

- Fixed in ROOT: crashes and glibc memory issues ([ROOT-5643](#))
- Fixed in ROOT: template parameter lookup diagnostic ([ROOT-5711](#))
- Fixed in PyROOT: “isinstance” checks in namespaces ([ROOT-5718](#))
- Fixed in PyROOT: cool.ValidityKeyMin lazy lookup ([ROOT-5746](#))
- Clarified, closed: ROOT_INCLUDE_PATH in LCGCMT ([ROOT-5623](#))
- Work-around “JIT does not support inline ASM” ([bug #102771](#))
 - Replace Boost by c++11 shared_ptr ([task #48846](#))... see next slides
 - A better workaround than using a patched Boost ([SPI-336](#))

Evolution in ROOT6 PyCool strategy...



- Friday: move COOL API from boost to c++11 std::shared_ptr
 - As a better workaround for JIT than the ASM-free Boost
 - But attempted this also for ROOT5 on COOL290... bound to fail!
- Monday: understood we need ROOT dictionaries only for I/O
 - After stumbling in another blocker for std::shared_ptr in dictionaries
 - First, try to load PyCool.h headers via ProcessLine and no .so
 - Then, aim instead at embedding PyCool.h in the .so without I/O info, using two selection.xml files on ROOT5 (boost) and ROOT6 (~empty)
- Tuesday: remove c++11 API from ROOT5-based COOL290
 - After re-discovering that ROOT5 does not support c++11...
- Wednesday: disable dictionaries via --interpreteronly
 - Better than empty selection.xml because it preserves rootmap?
 - Keep .so, two selection.xml files on ROOT5 (boost) and ROOT6 (c++11)
 - Add extra option only on ROOT6 (may try it on ROOT5 after LCG67...)

New issues found on the way

- Fixed in LCGCMT: clang builds do not use c++11 ([SPI-352](#))
- Medium: dict with enum template args ([ROOT-5757](#))
 - Was a blocker before adding workaround: use JIT, remove dictionaries from PyCool
 - Additional workaround: add `-fpermissive` to compile PyCool dict (except on clang)
- Low: segfault after failed ProcessLine ([ROOT-5759](#))
- Blocker: `std::smart_ptr` in `cppyy` ([ROOT-5761](#) – aka old [ROOT-5682?](#))
 - To be followed up (patches in ROOT and PyCool workaround proposed)
- Medium: rootcling on clang does not use c++11 ([ROOT-5765](#))
- Blocker: dict with namespaced template args ([ROOT-5766](#))
- High (SPI): unreliable nightlies using 1-week old COOL src ([SPI-356](#))
 - High (SPI), also: missing ssh access to some nightly nodes (old [SPI-323](#))
- *Plus several iterations and new bugs added to COOL, cleanup underway...*

Summary of ROOT6 pending issues in PyCool

- Blocker: dict with namespaced template args ([ROOT-5766](#))
- Blocker: std::smart_ptr in cppyy ([ROOT-5761](#) – aka [ROOT-5682?](#))
 - To be followed up (patches in ROOT and PyCool workaround proposed)
- High (old): c++ exception propagation to Python ([ROOT-5603](#))
- Medium: dict with enum template args ([ROOT-5757](#))
- Medium: rootcling on clang does not use c++11 ([ROOT-5765](#))
- Low: segfault after failed ProcessLine ([ROOT-5759](#))
- Low (old): assertions if LLVMDEV is enabled ([ROOT-5697](#))

- *We can now use the dev2 slot again (default Boost) to test PyCool!*
 - *No need for hacked Boost, as inline ASM issues have been bypassed!*

- *One PyCool build succeeded in the test slot at 3.10pm, several runtime failures of PyCool tests will be debugged and followed up...*

Thanks to the usual suspects for the help and suggestions!