



# **POOL Release V1.1**

---

**Dirk Düllmann**  
**LCG Application Area Meeting,**  
**9<sup>th</sup> July 2003**



# POOL V1.1 Release

---

- POOL V1.1 has been released on time
  - Public release targeting expert developers expecting to use POOL directly (eg experiment framework developers)
    - CVS release tag "POOL\_1\_1\_0"
    - Build against SEAL 0.3.2
  - Release notes at <http://pool.cern.ch/relnotes-1.1.0.html>
  - Please use savannah for feedback <http://savannah.cern.ch/projects/pool/>
- One supported platform so far
  - RedHat 7.3 using gcc 3.2
    - Preparations to extend the platform list for test builds underway in SPI
- External packages hosted by SPI in </afs/cern.ch/sw/lcg/external>
  - MySQL (4.0.4-beta)
  - MySQL++ (1.7.9)
  - ROOT (3.05.05)
  - Xerces-C (2.1.0)
  - Edg-rls-client (1.3.0)



# Storage Manager



- Several **significant** improvements including
  - Support for transient data members
    - Declared in the SEAL dictionary
  - Support for embedded C++ pointers
    - Also as part of STL collections
  - Support updates on the streaming layer
    - Needed to keep meta data inside ROOT
    - Implemented only for ROOT I/O directory technology
  - Simplified end-user transaction scheme
    - Removes need for user code to keep track of individual file connections and transactions
    - Allows also to subscribe experiment code to execute on transaction start, commit, rollback





# Implicit (Storage Manager) Collections

---

- New in V1.1
- Allow to iterate through all objects in a given POOL database
  - Interface compatible with the other collection implementations provided before
- Additional API provided to enumerate all POOL containers in a given database
- Important conclusion on the storage manager functionality for POOL
  - As for the File Catalog we now provide a comprehensive set of replaceable backend implementations for maintaining large object collections in a technology independent way



# ROOT Explicit Collections



- New package which allows to maintain POOL explicit collections in ROOT
  - Base on standard ROOT tree which keeps a POOL token as reference for eg an event in another file
  - Interface compliant to other POOL collections
- Presentation about the different collection implementations provided by POOL in an AMM soon.
  - Need to see how this high level facilities integrate with PI, the experiment frameworks and grid “collections”



# File Catalogs



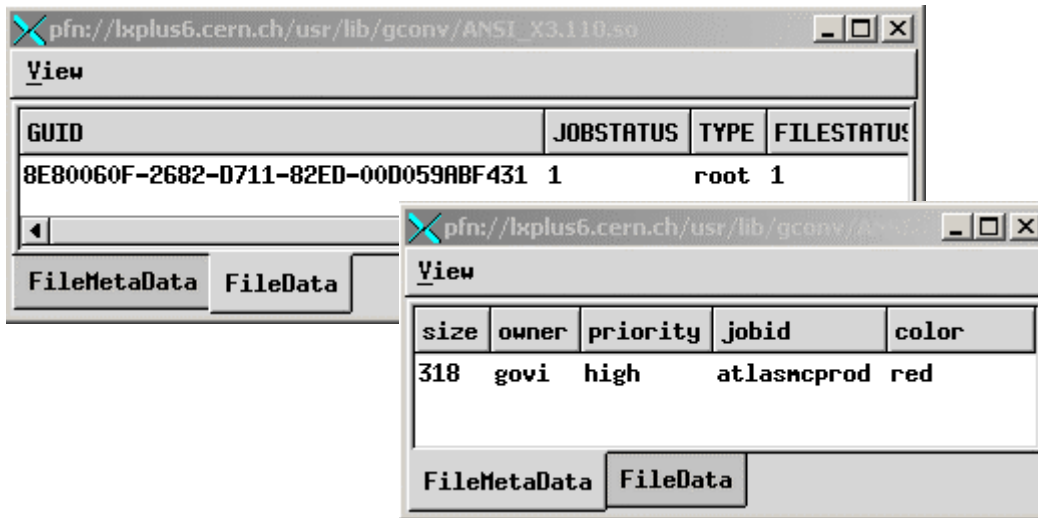
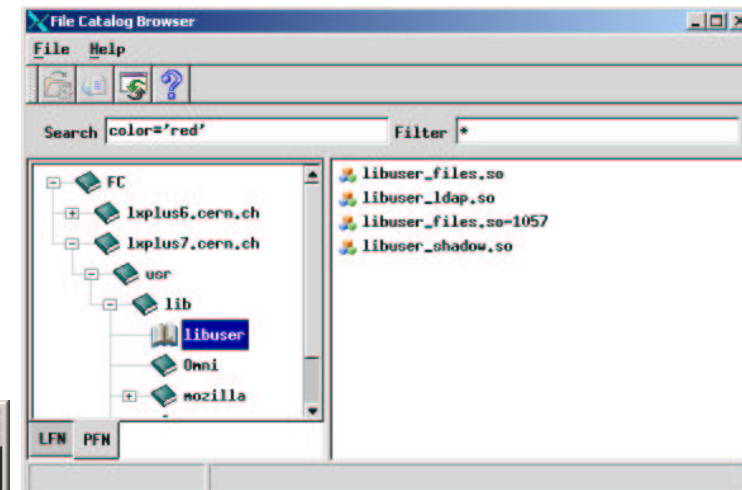
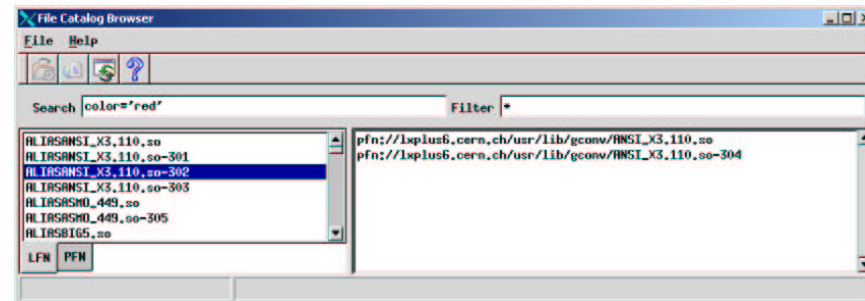
- Consolidation of the transaction protocol between all Catalog implementations
  - Now checking adherence to the protocol as part of the release tests
  - To insure that a change of catalog backend does not introduce semantic or even code changes in the client application
- New EDG client has been integrated
- Produced an estimate of the lookup frequencies
  - Based on experiment references for LCG-1 and 2008
- Starting to define the catalog backup and recovery models
  - Propose that the persistency project coordinates the gathering of requirements and produces a service description
  - Together with the experiments and the RLS service providers



# POOL File Catalog Browser Prototype



- File Catalog browser now part of the supported packages
- POOL Work Book entry in preparation
- Discussion on extension to a simple object browser started with SEAL



# Infrastructure and Testing



- POOL contains now 63 individual tests
  - We got several feature request backed up by an acceptance test from the requester. Thanks!
- All validation tests are now run automatically during the release build procedure
  - using the SPI provided Oval and CPPUnit
- Output of all tests is kept with each POOL release in AFS
- Performance (and scalability) tests are not fully automated and not executed during each release as they often require significant hardware and CPU resources and human interpretation of their respective results
  - Rather run by their package developers at least once per release cycle.





# POOL Release Scripts and Integration into the Nightly Builds



- POOL uses a set of scripts to fully automatise a release
  - We mostly care about our internal releases as they are
    - Much more frequent
    - Harder to configure (as typically not all packages are be build)
  - These scripts around scram are used as a stop gap solution until scram has been enhanced to provide
    - Automatic dependency calculation for the build and (!) test sequence
    - Partial POOL builds without having to change the scram configuration
- These scripts are now also used from the NICOS build system
  - Alex will provide feedback to ensure that these script do not need any modification for the nightly build environment



# EDG RLS Test Service



- Already with the POOL V1.0 release EDG WP2 has started to provide a RLS test service for POOL
  - based on EDG-RLS running on Oracle 9iAS and 9i database
  - Thanks to the WP2 RLS team
- Please consult
  - <http://edg-wp2.web.cern.ch/edg-wp2/replication/rls-for-pool.html> for details on this service
- POOL is now using this service for the internal validation tests
- This service will soon be replaced by LCG-1 production services dedicated to individual experiments



# In the queue...

---



- Automated dictionary loading
  - Will be provided by a high level Dictionary service in SEAL
- Remove the need to specify a Class ID property for each class in the dictionary steering file
- Allow to leave the object ownership to the experiment framework
  - Introduce as global policy - leave all object deletes to user code



# POOL Cache and Experiment Caches



- RTAG : provide a Ref but leave the cache responsibility to experiments
  - POOL needs at least one cache for validation and simple example programs
  - CMS is using this cache right now
- Started a discussion about the role of the POOL cache
  - Two approaches discussed with ATLAS
    - Integration at the (low) Persistency Service level (replacing the POOL provided cache by existing experiment code (eg StoreGate))
    - Integration on top of the POOL cache, allowing to customise the cache policy implemented by POOL
  - Need interested experiments validate the approaches with prototype integration
  - Also connected to the use of an object white board



# Next POOL Release Cycle

---



- The POOL V1.2 release cycle will address
  - Performance and Scalability testing
  - Creating the test plans by end of this week
    - Performance and Scalability tests for all components
    - Comparison with ROOT I/O in case of the storage manager
  - Otherwise prefer to leave the pipeline open for bug fixes requested by the experiment integration teams
- Initiated implementation review of POOL ROOT I/O storage service with Rene and Fons
  - will report back in a future AAM
- **Planned POOL V1.2 release date is 1st August**
  - **code freeze 25. July**



# Conditions Database Subproject

---



- SC2 has initiated a new sub-project
  - Clear priority to POOL project at this point
  - Proposed Andrea Valassi as sub-project leader
- Next steps
  - Produce a draft work plan for this sub-project
  - Identify the relevant conditions db implementations
  - Integration of those with SPI development infrastructure
  - Contact to people in the experiments who are willing to participate



# Summary

---



- POOL V1.1 is a very big step forward
  - Basically all requested functionality changes before the code freeze date have been implemented on time!
- We consider POOL V1.1 to be functionally ready for experiment integration
  - and we get promising reports about first integration successes
- **BIG thanks to all people on the POOL team and to EDG, SEAL, SPI and ROOT teams who support us!**

