



*GDB*

## “POOL to LFC interface”

David Smith, IT-GD, CERN

CERN, 20th July 2005





# Overview



- Provide the POOL file catalog with a way to interact with the LFC catalog
- The POOL-LFC interface in turn uses the LFC C client to communicate with the LFC server



# The POOL-LFC interface is



- The POOL-LFC interface:
- For file catalog functionality:
  - `connect()`, `disconnect()`, `start()`, `commit()`, `rollback()`,  
`registerPFN()`, `registerLFN()`, `addReplicaPFN()`,  
`addPFNtoGuid()`, `renamePFN()`, `lookupFileByPFN()`,  
`lookupFileByLFN()`, `lookupBestPFN()`, `insertPFN()`,  
`insertLFN()`, `deletePFN()`, `deleteLFN()`, `deleteEntry()`,  
`isReadOnly()`, `retrievePFN()`, `retrieveLFN()`, `retrieveGuid()`
- *Not implemented:*
  - `insertMetaData()`, `createMetaDataSpec()`,  
`updateMetaDataSpec()`, `getMetaDataSpec()`,  
`dropMetaDataSpec()`, `deleteMetaData()`, `retrieveMetaData()`



# Aim of the interface



- Allow POOL FC to provide useful service with LFC
  - Where possible allow existing code and command line tools to remain useful
- Operations should also perform *reasonably*
  - Complications to do this due to:
    - LFC uses hierarchical namespace for entries (i.e. LFNs)
    - The queries that POOL allow to be passed to the file catalog interfaces are not completely specified by POOL
    - LFC requires that a replica (i.e. PFN) is always associated with an LFN
    - Support symbolic links (i.e. aliases for LFNs)
    - POOL FC requires that all FC operations are done within a *transaction*
    - The natural use of the POOL FC interface and its command line tools could easily result in sub optimal LFC queries



# About Hierarchical LFNs



- LFNs are understood to be hierarchical
  - Relative LFNs are supported – they have a user specified prefix added to make them absolute
  - The same is true for query strings on LFNs
- Auto-generated LFNs
  - Are created in an name based on the file GUID
  - The location of the directory is set by a user defined location
- Both of the above specified in the *contact* string
  - `lfccatalog_lfc://host.cern.ch[:port][/relative/prefix[:/gen/path]]`



# Insert & removal of entries



- Insert of entries
  - To insert a new PFN the interface has to create an auto-generated LFN
  - Adding an LFN to a PFN with only an auto-generated LFN renames the generated LFN with the user specified one
  - Adding an LFN to an existing user LFN creates a symbolic link
- Removing entries
  - Removing a PFN relatively straightforward
  - Removing an LFN has to deal with several cases:
    - A symbolic link
    - The last LFN for zero, one or more PFNs
    - An LFN for which there are symbolic links



# Lookup and Queries



- Lookup of GUID from LFN or PFN is supported are a dedicated method
- Lookup of LFNs & PFNs are a sub class of *queries*
- Queries
  - Queries for lists of PFN, LFN or GUID are done with a *query string*
  - A POOL query string resembles an SQL fragment:
    - *lfname*, *pfname* and *guid*
    - =, *LIKE*
    - *AND*
    - Wildcards supported: % \_



# Form of Queries



- Queries are formed as:
  - pfname='srm://host.cern.ch/path'
  - lfname like myname\_
  - lfname like something% and pfname like %other%
- For the purposes of matching LFNs, directory elements are treated separately e.g.:
  - *lfname like /%* matches files in /, not the whole catalog
  - *lfname like /grid/%/%* matches all files in directories in /grid
- Queries that imply a scan over all the LFN space, such as:
  - pfname like %string%
- Are understood as:
  - pfname like %string% and lfname like %





# Uniqueness of query results



- By default results will be unique
  - Either by construction of the query
  - Otherwise by explicitly eliminating possible duplicates
    - Potentially extremely costly
    - Number of requests grows rapidly with total result size / available temporary store for the interface (i.e. memory)
  - Uniqueness check can be disabled



# Transactions



- The POOL FC requires that a catalog interface is *started* before catalog inserts or lookups are done and then *committed* or *rolled back*
  - The POOL-LFC interface starts transactions for the duration
  - However LFC queries that can return multiple results (such as lists of PFNs or LFNs) are done via a database cursor but outside of the transaction
    - Therefore for a given transaction the POOL-LFC catalog maintains a list of changed entries to allow lookups to return entries changed during an as yet uncommitted transaction
    - However there still exists the possibility that cursor and transaction could see a different database view



# Summary



- POOL-LFC a part of POOL since POOL release 2.1.0
  - The interface as described here is present in release 2.1.1
- POOL-LFC interface under testing by EIS (mainly using examples inspired from CMS use cases)
  - Several suggestions made and problems found
- Known issues to address
  - Performance of selections on or long listings of PFNs
  - Possible problems due to readdir() outside of transaction
- but mainly
  - Making sure the interface is as useful as possible to those who want to use it...