Using CORAL

This page contains our experience with using CORAL. My conclusions so far:

- it is nice, well thought-through code
- some essential features (like loading via csv files) are missing
- the code provides the right level of dbms isolation from LSST point of view

Note: CORAL requires mysql to be started in ansi mode. This means eg. adding a line "sql-mode = ansi" to my.cnf

---

There is a problem with types on a 64-bit architecture: for a table with column defined as INT, I was able to write data using INT attribute, but when I try to read it back I am getting:

```
C++ Exception : Attempt to assign attribute "id" of type long with int ( CORAL : "Attribute" from "CoralBase
```

This is because mysql returns FIELD_TYPE_LONG type for columns defined as INT, and CORAL incorrectly assigns it to the C++ long type. db/mysql/MySQLExample01 shows how to reproduce it. I fixed it in CORAL/CORAL_1_7_0/src/MySQLAccess/src/Statement.cpp

---

I noticed that CORAL always refers to tables using databaseName.tableName syntax. This has certain disadvantages.

CORAL uses AttributeList class to represent data for a row. AttributeList keeps data as std::vector<Attribute>, there is a way to get to data address for each Attribute. From our point of view it would be better to have a single address space for entire row (and I don't think we want to expose AttributeList to Formatter). I guess this would make type checking more difficult, have to think about it more...

This will likely be the biggest bottleneck to large-volume usage of Coral -- the packing and unpacking of data into and out of AttributeLists. Adding another layer on top to hide the AttributeList from the Formatters is likely to just add even more memory copies and overhead. -- KTL

Issue: transactions are tightly bundled with sessions, I don't see how we could have multiple sessions inside a single transaction.

Within a single database (and thus session), multiple tables can be accessed within one transaction. Any transaction semantics that span databases (or involve non-database entities like files) will need to be handled by the application anyway. I don't see this as being much of an issue. -- KTL

More coming soon...