

Valeriy Onuchin

ROOT Database Connectivity



Valeriy Onuchin, IHEP, Protvino

MINOS

Hardware:

- **Near Detector (ND)**: at Fermilab
- **Far Detector (FD)**: in northern Minnesota
- **Calibration Detector (CalDet)**: portable, presently at CERN test beam

Database architecture:

- **Central Warehouse**: This is currently thought to be **Oracle** at FNAL.
- **Detector DBs**: Currently thought to be **Oracle** as well at **ND** and **FD**, **MySQL** at **CalDet**.
- **Institutional DBs**: Collaborating institutions may want to have their own database servers, rather than constantly accessing Central.
These are expected to be mainly **MySQL**-based, with subsets of the data from Central.

Database Interface, or DBI:

- **MINOS Database Interface** is developed by Pete Border, Nick West and me and used for extracting the large quantities of constants needed for reconstruction.
- **RDBC** or **ROOT Database Connectivity** as part of **DBI** is the lowest level API which provides uniform interface to Oracle, MySQL as well as to db tables stored in ASCII files.



Features

- **RDBC aims for JDBC2.0 "compliance":**
 - `TSQLDriverManager`, `TSQLConnection`, `TSQLStatement`, `TSQLPreparedStatement`, `TSQLCallbleStatement`, `TSQLResultSet`, `TSQLResultSetMetadata`, `TSQLDatabaseMetadata` classes have their "twins" in JDBC
- **RDBC aims for ROOT SQL compliance:**
 - `TSQLResultSet` is subclass of ROOT SQL's `TSQLResult`. Internal `RDBCRow` class is subclass of `TSQLRow`.
 - `TRDBCServer` is subclass of ROOT SQL's `TSQLServer` class.
 - In general, program written with ROOT SQL classes should work with RDBC. RDBC can be considered as an extension of ROOT SQL interface.
- **RDBC separated into interface (ABCs) and implementation parts.**
 - Implementation part translates RDBC calls into ODBC operations. That allows to connect and work with any database for which **ODBC driver** available.
 - ODBC implementation is based on `libodbc++` library (<http://orcane.net/freeodbc++/>) developed by **Manush Dodunekov <manush@stendahls.net>**



Features++

- **URL = dynamic DSN**
 - RDBC allows to use JDBC like format of connect string. **Comment:** connecting via ODBC requires DSN (*DataSourceName*).
RDBC creates and writes DSN into \$ODBCINI file "on-the-fly" by parsing of URL connect string.
- **Exceptions handling via Signal-Slot communication mechanism**
 - Signal "Throw(TSQLEException*)" is emitted in case of error. It can be connected to handler function.
- **"Connections" to tables stored in remote/local ASCII files**
 - RDBC allows to dynamically load database tables stored in ASCII file written in CSV format
- **Connections pooling**
 - TSQLConnections are ref.counted and reusable during session.
- **Persistence for TObject**
 - TSQLResultSet:GetObject, TSQLResultSet:UpdateObject, TSQLPreparedStatement:SetObject methods allow to write/read ROOT objects to/from SQL database.
- **RDBC and ROOT-Apache module**
 - RDBC is supposed to be used with **ROOT-Apache Module** (check my next talk about it). Some Apache specific features like "persistent connections" will appear soon.



Example

```
// Set exception handler
TSQL::SetHandler("Catch(TSQLException*)");

// Open a connection...
TSQLConnection* con = TSQLDriverManager::GetConnection("mysql://localhost/test", "myLogin", "myPassword");

if(!con) return;

// Create a statement...
TSQLStatement* stmt = con->CreateStatement();

// Execute the query... Get the result set...
TSQLResultSet* rs = stmt->ExecuteQuery("SELECT a, b, c FROM Table1");

if(!rs) return;

while(rs->Next()) { // Advance to the next row...

    // Get the data...
    Int_t x = rs->GetInt("a");
    TString s = rs->GetString("b");
    Float_t f = rs->GetFloat("c");
}

}
```



TODO

Short-term period (1–2 months)

- **RDBC–MySQL "thin" driver**
 - RDBC API implementation with MySQL C library
- **"Persistent" TSQLConnections**
 - RDBC connections allocated in shared memory

Long-term period

- **Testing against other OpenSource Databases:**
 - PostgreSQL, SAP DB, ODBC–ODBC bridge etc.
 - benchmark tests
- **Porting to Windows**
- **RDBC–Oracle OCI8 driver (??)**
 - RDBC API implementation with OCI8 library

