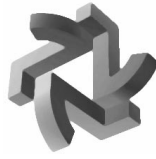


**Valeriy Onuchin**

# **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++

- **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(TSQLError\*)" 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

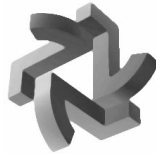
```
// Set exception handler
try {
    SQLConnection con = DriverManager.getConnection("mysql://localhost/test", "myLogin", "myPassword");
} catch (SQLException e) {
    e.printStackTrace();
}

// Open a connection...
SQLConnection con = DriverManager.getConnection("mysql://localhost/test", "myLogin", "myPassword");
if (!con.isClosed()) {
    // Create a statement...
    SQLStatement stmt = con.createStatement();

    // Execute the query... Get the result set..
    SQLResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");

    if (!rs.isClosed()) {
        while (rs.next()) {
            // Advance to the next row...

            // Get the data...
            int x = rs.getInt("a");
            TSString s = rs.getString("b");
            float 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

