



USING MONGODB WITH
SOURCEPRO DB

INTRODUCTION TO MONGODB

MongoDB is a document-based database whose document-oriented model allows it to automatically split data across multiple servers. MongoDB is schema-free with no predefined or fixed document keys. Some features common to relational databases are not present in MongoDB — notably joins and complex multi-row transactions¹.

USING MONGODB WITH SOURCEPRO DB

To perform queries on a MongoDB database, use the SourcePro DB ODBC Access Module and Apache Drill, an open source framework inspired by Google's Dremel. Drill allows access to nested data attributes as if they were SQL columns².

The example here uses a Drill plugin to connect to MongoDB. Apache Drill is supported on Windows and Linux. Here, we demonstrate how to set up and run an example on Windows.

Installing and configuring MongoDB Client and Apache Drill

1. Install the MongoDB Client.
 - Download MongoDB from <https://www.mongodb.org/downloads>
 - Run the installer and choose "custom installation," then select the client component to be installed.
2. Install Apache Drill in embedded mode. Embedded mode installs Drill locally and requires no configuration.
 - Ensure that the following prerequisites are met for running Apache Drill:
<https://drill.apache.org/docs/embedded-mode-prerequisites/>
 - Download and install Apache Drill on Windows: <https://drill.apache.org/docs/installing-drill-on-windows/>
3. Install Apache Drill ODBC driver.
 - Download the Apache Drill ODBC driver from step 1 at:
<https://drill.apache.org/docs/installing-the-driver-on-windows/>
 - Install and configure a DSN with the Apache Drill ODBC driver using the procedure in step 2 at:
<https://drill.apache.org/docs/installing-the-driver-on-windows/>
4. Start Apache Drill.
 - Start the drill shell using the sqlline command:

```
sqlline.bat -u "jdbc:drill:zk=local"
```

Note: If you run into errors starting sqlline command, ensure that JAVA_HOME is set and the path doesn't have double quotes.

- Enable the MongoDB plugin provided by Apache Drill by navigating to your local host at <http://localhost:8047/storage> and clicking on "Enable Mongo."
- Verify that your MongoDB database is detected by Apache Drill using the sqlline command show databases. Here is a sample output:

¹ "MongoDB: The Definitive Guide", Kristina Chodorow and Michael Dirolf, May 2013.

² "Apache Drill: Architecture Introduction", drill.apache.org, accessed May 2016.

```

0: jdbc:drill:zk=local> show databases;
+-----+
|      SCHEMA_NAME      |
+-----+
| INFORMATION_SCHEMA    |
| cp.default            |
| dfs.default           |
| dfs.root              |
| dfs.tmp               |
| mongo.admin           |
| mongo.employee        |
| mongo.local           |
| mongo.test            |
| sys                   |
+-----+

```

5. Import the zip database.

- Download zip.json from the Apache Drill website <http://media.mongodb.org/zips.json#>
- Import zips.json into your MongoDB server.

For this example, the json file was imported into a collection zips inside the database test.

```

C:\Program Files\MongoDB\Server\3.0\bin> mongoimport --host localhost --db test
--collection zips < C:\zips.json
2015-09-16T16:51:40.714-0600    connected to: localhost
2015-09-16T16:51:41.284-0600    imported 29353 documents

```

Example on querying MongoDB/Apache Drill using SourcePro DB

This example demonstrates reading data from a MongoDB collection using SourcePro DB, executing the following query:

```

SELECT sum(pop), city
FROM mongo.test.zips
WHERE state = 'CO' and pop > 1000
GROUP BY city
ORDER BY sum(pop) DESC

```

Note that a select query can be executed similarly to the way in which you would query a relational database. You can't, however, write to a MongoDB database using Apache Drill as Drill currently provides read-only support³.

³ "Apache Drill: MongoDB Plugin for Apache Drill", drill.apache.org, accessed May 2016.

```

#include <iostream>
#include <rw/db/db.h>

void selectFromZipsTbl(const RWDBConnection&, const RWDBTable&);

int main() {
    // Connect to MongoDB using configured DSN
    RWDBDatabase db = RWDBManager::database("ODBC", "<MONGODB DSN NAME>", "<USER>",
"<PASSWORD>", "");
    RWDBConnection conn = db.connection();
    RWDBTable zipsTbl = db.table("mongo.test.zips");
    selectFromZipsTbl(conn, zipsTbl);
    return 0;
}

void selectFromZipsTbl(const RWDBConnection& conn, const RWDBTable& zipsTbl) {
    RWDBSelector sel = conn.database().selector();
    sel << rwdbSum(zipsTbl["pop"]);
    sel << zipsTbl["city"];
    sel.where((zipsTbl["state"] == "CO") && (zipsTbl["pop"] > 1000));
    sel.groupBy(zipsTbl["city"]);
    sel.orderByDescending(rwdbSum(zipsTbl["pop"]));
    RWDBTable table = sel.execute(conn).table();

    // Read data returned from select query
    RWDBReader rdr = table.reader(conn);
    int sum;
    RWCString city;
    while (rdr()) {
        rdr >> sum >> city;
        std::cout << sum << "\t" << city << std::endl;
    }
}

```



Rogue Wave provides software development tools for mission-critical applications. Our trusted solutions address the growing complexity of building great software and accelerates the value gained from code across the enterprise. The Rogue Wave portfolio of complementary, cross-platform tools helps developers quickly build applications for strategic software initiatives. With Rogue Wave, customers improve software quality and ensure code integrity, while shortening development cycle times.