

# Writing a new DataStore for Gora HOW\_TO

## Writing a New DataStore for Apache Gora

### What You Need

A couple of hours time and some patience...

### Step 1 - Get the Gora Code

Check out the Gora trunk source code from one of the many sources, the documentation on how to do this is located [here](#).

### Step 2 - Create a New Module

Depending on how you obtained the Gora trunk source code above, there are numerous possible methods for how you do this. The best advice is to copy another module (such as gora-accumulo) to use as a template.

In SVN

```
svn cp gora-accumulo gora-$moduleName
```

In Git

```
cp gora-accumulo gora-$moduleName  
git add gora-$moduleName
```

**N.B** Beware the curse of SVN here when copying can sometimes get you in to a bit of a mess.

You will now have the full directory structure which will look something similar to this:

```

.
├── lib
├── pom.xml
├── src
│   ├── examples
│   │   └── java
│   ├── main
│   │   └── java
│   │       ├── org
│   │       │   ├── apache
│   │       │   │   └── gora
│   │       │       └── accumulo
│   │       │           ├── encoders
│   │       │           │   ├── BinaryEncoder.java
│   │       │           │   ├── Encoder.java
│   │       │           │   ├── HexEncoder.java
│   │       │           │   ├── SignedBinaryEncoder.java
│   │       │           │   └── Utils.java
│   │       │           ├── query
│   │       │           │   ├── AccumuloQuery.java
│   │       │           │   └── AccumuloResult.java
│   │       │           ├── store
│   │       │           │   ├── AccumuloMapping.java
│   │       │           │   └── AccumuloStore.java
│   │       │           └── util
│   │       │               └── FixedByteArrayOutputStream.java
│   └── test
│       ├── java
│       │   ├── org
│       │   │   ├── apache
│       │   │   │   └── gora
│       │   │       └── accumulo
│       │   │           ├── store
│       │   │           │   ├── AccumuloStoreTest.java
│       │   │           │   └── PartitionTest.java
│       │   │           └── util
│       │   │               ├── HexEncoderTest.java
│       │   │               └── SignedBinaryEncoderTest.java
│       └── resources
│           ├── gora-accumulo-mapping.xml
│           └── gora.properties

```

23 directories, 17 files

## Customize the Directory Structure

Here we wish to replace **ALL** instances of the old module name with the new one. For this we can use a tool such as [sed](#) which makes life much easier. The following command can be run from within the **new** module directory.

The first command recursively replaces all occurrences of the term *Accumulo* (uppercase **A**) with *\$moduleName*

```
sed -i 's/Accumulo/$moduleName/g' `grep -rl 'Accumulo' *`
```

The second command recursively replaces all occurrences of the term *accumulo* (lowercase **A**) with *\$moduleName*

```
sed -i 's/accumulo/$moduleName/g' `grep -rl 'accumulo' *`
```

## Establish Maven Parent pom.xml Configuration

Navigate to the Gora parent pom.xml. Find the `<modules></modules>` section of the pom tree and add your new modules there

```
<modules>
  <module>gora-core</module>
  <module>gora-hbase</module>
  <module>gora-accumulo</module>
  <module>gora-cassandra</module>
  <!-- module>gora-solr</module-->
  <module>gora-dynamodb</module>
  <!--module>gora-sql</module-->
  <module>gora-tutorial</module>
  <module>sources-dist</module>
</modules>
```

Then locate and add the version of the new datastore and any other properties to the `<properties></properties>`. This merely acts as a central convenient location to house the datastore specific version properties.

```
<properties>
  <!-- Dependencies -->
  <osgi.version>4.2.0</osgi.version>
  <!-- Avro Dependencies -->
  <jackson.version>1.6.9</jackson.version>
  <!-- Hadoop Dependencies -->
  <hadoop.version>1.0.1</hadoop.version>
  <hadoop.test.version>1.0.1</hadoop.test.version>
  <hbase.version>0.90.4</hbase.version>
  <avro.version>1.3.3</avro.version>
  <cxfrt.frontend.jaxrs.version>2.5.2</cxfrt.frontend.jaxrs.version>
  <!-- Amazon Dependencies -->
  <amazon.version>1.3.12</amazon.version>
```

Finally navigate to the `<dependencyManagement></dependencyManagement>` section of the pom.xml and add in the new modules.

```
<dependencyManagement>
  <dependencies>
    <!-- Internal Dependencies -->
    <dependency>
      <groupId>org.apache.gora</groupId>
      <artifactId>gora-core</artifactId>
      <version>${project.version}</version>
    </dependency>
    <dependency>
      <groupId>org.apache.gora</groupId>
      <artifactId>gora-core</artifactId>
      <version>${project.version}</version>
      <type>test-jar</type>
    </dependency>
```

## Establish Maven Module pom.xml Configuration

Open the new project pom.xml and replace all old dependencies with new ones.

## Replace Old Files Names With New

This simply consists of manually replacing (may require svn mv or something similar) all of the old Java and configuration file names with the new ones. You can hopefully copy, edit and paste the following string which makes the replacements and can be run from the terminal.

```
mv src/main/java/org/apache/gora/accumulo/ mv src/main/java/org/apache/gora/$moduleName/  
mv src/test/java/org/apache/gora/accumulo/ mv src/test/java/org/apache/gora/$moduleName/  
mv src/main/java/org/apache/gora/voldemort/query/accumuloresult.java src/main/java/org/apache/gora/$moduleName/  
/query/$moduleNameResult.java  
mv src/main/java/org/apache/gora/voldemort/query/accumuloquery.java src/main/java/org/apache/gora/$moduleName/  
/query/$moduleNameQuery.java  
mv src/main/java/org/apache/gora/voldemort/store/accumuloStore.java src/main/java/org/apache/gora/$moduleName/  
/store/$moduleNameStore.java  
mv src/main/java/org/apache/gora/voldemort/store/accumuloMapping.java src/main/java/org/apache/gora/$moduleName/  
/store/$moduleNameMapping.java  
mv src/test/resources/gora-accumulo-mapping.xml src/test/resources/gora-$moduleName-mapping.xml  
mv src/test/java/org/apache/gora/voldemort/store/accumuloStoreTest.java src/main/java/org/apache/gora/$moduleName/  
/store/$moduleNameStoreTest.java
```

... of course there may be others but hopefully this is a help.