

Setting up Sqoop 1

In order to setup your development environment, you would need a Linux system with administrative privileges and Internet connection such as Ubuntu or CentOS. You would also need sufficient disk space for checking out and building the code, installing various database/other software that you may need for your testing.

- [Getting ready to build](#)
- [Building the Sources](#)
- [Running Tests](#)
 - [Running unit tests](#)
 - [Create third-party lib directory](#)
 - [Setting up and running third-party tests](#)
 - [Setting up MySQL](#)
 - [Setting up PostgreSQL](#)
 - [Setting up Oracle](#)
 - [Running third-party tests](#)
 - [Setting up and running manual tests](#)
 - [Setting up SQL Server](#)
 - [Setting up DB2 Server](#)
 - [Running manual tests](#)
- [Building documentation](#)
- [Building tar-ball](#)

Please see the docs for the latest release of Sqoop 1 <http://sqoop.apache.org/docs/>. Some of the information below might be outdated

Getting ready to build

Once you have your Linux system ready with sufficient disk space and Internet connection, go ahead and install the following software:

- Git to checkout the source code
- The recent update of JDK 1.6
- Recent version of make
- Asciidoc version 8.6 or above
- Apache Ant 1.7 or above
- Findbugs version 1.3.9 or above
- Latest Eclipse IDE (or your IDE/Editor of choice)

Building the Sources

To get the source code, checkout the subversion "trunk" using the following command:

```
$ git clone https://git-wip-us.apache.org/repos/asf/sqoop.git
```

Once you have the code, you can build it by the following command:

```
$ cd sqoop
$ ant jar-all
```

You can use the `clean` target to delete previously built files from the workspace and run `jar-all` again to do a fresh build.

To see a list of all available targets that are available in the build type the following command:

```
$ ant -p
```

If you prefer working in Eclipse, you can generate the necessary project definitions as follows:

```
ant eclipse
```

Once these definitions are generated, you can import them in Eclipse as an existing project.

Running Tests

Running unit tests

Sqoop source code contains many unit tests that exercise its functionality. These tests can be run simply by using the following command:

```
ant test
```

Create third-party lib directory

Create a directory somewhere convenient on your development system. This directory will hold all the JDBC drivers that the tests will use. Once created, create (or edit) the `build.properties` file in Sqoop workspace root directory and set the the full path of this directory as the value of the property `sqoop.thirdparty.lib.dir`. For example:

```
sqoop.thirdparty.lib.dir=/opt/ws/3rd-party-lib
```

Setting up and running third-party tests

Third-party tests are end-to-end integration tests that exercise the basic Sqoop functionality against third-party databases. You should run these tests in order to rule out regression when testing any changes to the core system. Before you run these tests, you must setup the following databases:

Setting up MySQL

- Install MySQL version 5.1.x with necessary client tools. You can install the server in a different host than your development host if necessary. However, you must have the client tools available on your development host including the JDBC driver, and batch utilities such as `mysqldump` and `mysqlimport`.
- Place the JDBC driver in the third-party lib directory that you created earlier.
- The location of MySQL server is specified in the `build.properties` file by the value for the property `sqoop.test.mysql.connectstring.host_url`. This property defaults to `jdbc:mysql://localhost/` which assumes local installation and default port setup. If however your MySQL server is installed on a different host or on a different port you should specify it explicitly as follows:

```
sqoop.test.mysql.connectstring.host_url=jdbc:mysql://<mysqlhost>:<port>/
```

- In order to run the MySQL third-party tests, you would need to configure the database as follows:

```
$ mysql -u root -p
mysql> CREATE DATABASE sqoopasstest;
mysql> CREATE DATABASE sqooptestdb;
mysql> use mysql;
mysql> GRANT ALL PRIVILEGES on sqoopasstest.* TO 'sqooptest'@'localhost' IDENTIFIED BY '12345';
mysql> GRANT ALL PRIVILEGES ON sqooptestdb.* TO 'yourusername'@'localhost';
mysql> flush privileges;
mysql> \q
```

- **Note:**
 - If the installation of MySQL server is on a different host, you must replace the `localhost` with the appropriate client host value.
 - You should replace `yourusername` with your actual user name before issuing the command.

Setting up PostgreSQL

- Install PostgreSQL 8.3.9 or later along with client tools. You can install the server in a different host than your development host if necessary. However, you must have the client tools available on your development host including the JDBC driver and command line utility `psql`.
- Place the JDBC driver in the third-party lib directory that you created earlier.
- The location of PostgreSQL server is specified in the `build.properties` file by the value for the property `sqoop.test.postgresql.connectstring.host_url`. This property defaults to `jdbc:postgresql://localhost/` which assumes local installation and default port setup. If however your PostgreSQL server is installed on a different host or on a different port you should specify it explicitly as follows:

```
sqoop.test.postgresql.connectstring.host_url=jdbc:postgresql://<pgsqlhost>:<pgsqlport>/
```

- In order to run PostgreSQL third-party tests, you would need to configure the database as follows:
 - Edit the `pg_hba.conf` file and setup the authentication scheme to allow for testing. In a secured environment, it may be easy to setup up full trust based access by adding the following lines in this file, and commenting out any other lines referencing `127.0.0.1` or `::1`.

```
local all all trust
host all all 127.0.0.1/32 trust
host all all ::1/128 trust
```

- Also in the file `postgresql.conf` uncomment the line that starts with `listen_addresses` and set its value to `*` as follows:

```
listen_addresses = '*'
```

- Restart your PostgreSQL server after modifying the configuration files above.
- Create the necessary user and database for Sqoop testing as follows:

```
$ sudo -u postgres psql -U postgres templatel
templatel=> CREATE USER sqooptest;
templatel=> CREATE DATABASE sqooptest;
templatel=> GRANT ALL ON DATABASE sqooptest TO sqooptest;
tempalatel=> \q
$
```

Setting up Oracle

- Install Oracle 10.2.x or later and download the corresponding JDBC driver.
- Place the JDBC driver in the third-party lib directory that you created earlier.
- The location of Oracle server is specified in the `build.properties` file by the value for the property `sqoop.test.oracle.connectstring`. This property defaults to `jdbc:oracle:thin:@//localhost/xe` which assumes local installation and default port setup. If however your Oracle server is installed on a different host or on a different port you should specify it explicitly as follows:

```
sqoop.test.oracle.connectstring=jdbc:oracle:thin:@/<oraclehost>:<port>/<sid>
```

- In order to run Oracle third-party tests, you would need to configure the database as follows:

```
$ sqlplus system/<password>@<sid>
SQL> CREATE USER SQOOPTEST identified by 12345;
SQL> GRANT CONNECT, RESOURCE to SQOOPTEST;
SQL> CREATE USER SQOOPTEST2 identified by ABCDEF;
SQL> GRANT CONNECT, RESOURCE to SQOOPTEST2;
SQL> exit
$
```

- Note: If you are using Oracle XE and see an error like `ORA-12516, TNS:listener could not find available handler with matching protocol stack`, you are likely running into connection exhaustion problem. To circumvent this, log into the Oracle server as `SYSTEM`, run the command below and restart your server.

```
$ sqlplus system/<password>@<sid>
SQL> ALTER SYSTEM SET processes=200 scope=spfile;
SQL> exit
$
```

Running third-party tests

Once you have installed and configured all the above databases - MySQL, PostgreSQL and Oracle, you are now ready to run the third-party tests. To run them issue the following command:

```
$ ant test -Dthirdparty=true
```

Setting up and running manual tests

Certain third-party tests are categorized as Manual tests since these were introduced at a later stage and adding them to the third-party suite of tests would have resulted in ever test environment requiring new database installation.

Setting up SQL Server

- Install SQL Server Express 2008 R2 or above.
- Download and place the JDBC driver in the third-party lib directory that you created earlier.
- The location of SQL server is specified in the `build.properties` file by the value for the property `sqoop.test.sqlserver.connectstring.host_url`. This property defaults to `jdbc:sqlserver://sqlserverhost:1433` which assumes installation on a host called `sqlserverhost` and port 1433 setup. If however your SQL server is installed on a different host or on a different port you should specify it explicitly as follows:

```
sqoop.test.sqlserver.connectstring.host_url=jdbc:sqlserver://<sqlserverhost>:<port>
```

- In order to run SQL server manual tests, you would need to configure the database as follows:
 - Create a database called `SQOOPTEST`.
 - Create a login with name `SQOOPUSER` and password `PASSWORD`.
 - Grant all access for database `SQOOPTEST` to the login `SQOOPUSER`.

Setting up DB2 Server

- Install DB2 9.74 Express C.
- Download and place the JDBC driver in the third-party lib directory that you created earlier.
- The location of DB2 server is specified in the `build.properties` file by the value for the property `sqoop.test.db2.connectstring.host_url`. This property defaults to `jdbc:db2://db2host:50000` which assumes installation on a host called `db2host` and port `50000` setup. If however your DB2 server is installed on a different host or on a different port you should specify it explicitly as follows:

```
sqoop.test.db2.connectstring.host_url=jdbc:db2://<db2host>:<port>
```

- In order to run DB2 server manual tests, you would need to configure the database as follows:
 - Create a database called `SQOOP`.
 - Create a username `sqoop` with password `PASSWORD`.
 - Grant all access for database `SQOOP` to login `sqoop`.

The default username, password and database can be also overridden using following properties:

- `sqoop.test.db2.connectstring.database` for database
- `sqoop.test.db2.connectstring.username` for username
- `sqoop.test.db2.connectstring.password` for password

Running manual tests

Once you have installed and configured all the above databases - SQL Server and DB2, you are now ready to run the manual tests. To run them, issue the following command:

```
$ ant test -Dmanual=true
```

Building documentation

To build Sqoop documentation, run the following command from the workspace root directory:

```
$ ant docs
```

This will generate the documentation in the directory `build/docs` directory. To see the documentation, open the file `build/docs/index.html` in a web browser, where you will find the links to user and developer guides. All the man pages that are generated by this are available directly under `build/docs` directory with the extension `<name>.1.gz`. You can look at these man pages without installing them by the following command:

```
$ man -l sqoop.1.gz
```

Building tar-ball

To build the tar-ball for distribution, use the following command:

```
$ ant tar
```

This will produce a tar-ball distribution file with a name `sqoop-<version>.tar.gz` under the build directory.