

Sensor Data Access for Rasdaman

This project connects Rasdaman as a datasource backend to the [52°North SOS](#). The initial GSoC project idea was refined and now, a short description of the projects is:

- The [ASQLDB](#) is connected as a datasource backend to the [52°North SOS](#)
- The ASQLDB uses Rasdaman in order to store the columns that have the Array type
- ASQLDB was developed starting from the HSQLDB project, so the query language is similar with that one, the only difference being introduced by the Array columns. Details about the ASQLDB query language can be found in the project's [README](#) file.

A detailed development report for the Sensor Data Access for Rasdaman project can be found on the [52°North Wiki](#).

An overview of the project is presented in three blog posts created at the beginning, the middle and the end of the project:

- [First blog post](#)
- [Midterm blog post](#)
- Final blog post

Install ASQLDB in the local Maven repository

You can't skip this step even though you can [find ASQLDB in Maven Central](#).

Download the [zip file](#) representing the ASQLDB installation folder in the local Maven repository.

Unzip the archive and place it in: `.m2/repository/org/asqldb`

The `.m2` folder can be found in `~/.m2` for *Unix/Mac OS X* users. The Windows users can find this directory at `C:\Documents and Settings\{your-username}\.m2`

Install Rasdaman

Rasdaman can be installed only on Linux.

All the installation details can be found on the organization's page: [Rasdaman quick installation guide for Debian](#). You can ignore the versions of the packages that are suggested in the installation guide because usually you should install the latest version.

This guide should be sufficient for an user that is used with Linux. For a less experienced user, I've written a [more detailed installation guide](#) that lists almost each command that should be executed in order to finish the installation.

Install the 52°North SOS

In order to use the SOS webapp, you need to deploy the SOS on a local server - [Apache Tomcat](#) for example. Here is the [SOS installation guide](#) which describes two options for deploying the SOS on a local server:

- Build the SOS webapp and deploy it on a local server

- Download the war file and deploy it on a local server - **not available for the version of the SOS that has connects Rasdaman as a datasource backend.**

The SOS source code that connects Rasdaman as a datasource backend can be found on [GitHub](#), the develop branch.

Building the SOS project([the version that connects Rasdaman as a datasource](#))

It can be done in two different ways:

1. Build without running Rasdaman datasource integration tests: ***mvn package***
2. Build and run Rasdaman datasource integration tests at the same time: ***mvn package verify -P integration-test***

Build ASQLDB

In order to use the ASQLDB's GUI you have to build it from source. This is not difficult at all and you should try it without hesitation. All you have to do is

1. clone this repository: <https://github.com/misev/asqldb>
2. go to asqldb/build
3. execute "ant hsqldb" command
4. if the build is successful, then you can execute "ant run" command and the hsqldb GUI will open

Configure Rasdaman datasource using the SOS webapp

After building and deploying the SOS on local Tomcat, you can start to configure the datasource - Rasdaman datasource in this case. Choose it from the list and you will see a list of configuration fields. If the database is not created yet, you can choose what username and password you wish, and the database will be created assigning the chosen credentials with it. Press *Next*, choose the admin username and password and then press *Install*. In this last step, all the necessary tables are going to be created in your database.

Batch example

Now that you have configured the database you should try to insert something in these tables the SOS has just created. Usually, this action can be done from the Test Client web interface, but the Rasdaman datasource doesn't support this operation yet, so you can populate the tables by running the RasdamanTestData class. This class can be found in the SOS project, hibernate-datasource-rasdaman module, in src/test/java/org/n52/sos/ds/datasource.

Remote debug using Eclipse

Demonstration