

DSpace with Oracle DB test instance

This page briefly describes how to set up a DSpace test instance that runs with the Oracle DB. Note that this method is **not suitable for production environments**; rather it is intended for committers and other community members who don't normally have access to an Oracle database but wish to test patches and other code relevant to DSpace with Oracle.

- Using a Virtual Machine
 - Install Virtualbox
 - Download and import virtual machine image
 - Initialize virtual machine
 - Set up the database
 - Run DSpace Remotely OR Install DSpace on Virtual Machine
 - Run DSpace Remotely (recommended)
 - Install DSpace on Oracle Virtual Machine
 - Install build tools
 - Build + install DSpace
 - Install Tomcat
- Using a Docker Container
 - Starting the DSpace Oracle database container
 - Configure the DSpace database connection
 - Double check the database server is running and start Tomcat
 - Stop, restart or remove the database server.

Using a Virtual Machine

Install Virtualbox

You will need Oracle VM Virtualbox. Binaries for all major operating systems are available free of charge from <https://www.virtualbox.org/>.

Download and import virtual machine image

Oracle makes available virtual machine images for developers. You need to register for a free Oracle Tech Network account and agree to various license agreements.

Download the Database App Development VM image from <http://www.oracle.com/technetwork/database/enterprise-edition/databaseappdev-vm-161299.html> and import it into virtualbox as a new appliance. I increased the memory for the virtual machine to 3 GB.

Initialize virtual machine

Start the virtual machine (as per instructions on the download page, use username+password oracle).

Set up the database

1. Open a terminal and log on to the database as a superuser (password: oracle):

```
sqlplus logon AS SYSDBA

# NOTE: If you get at "invalid username/password; logon denied" error, try THIS COMMAND INSTEAD:
sqlplus 'sys as sysdba'
```

2. Then create the database user and database (which is referred to as a "schema" in oracle terms): (Thanks to [Hardy Pottinger](#) for the SQL statements.)

```
CREATE USER dspace IDENTIFIED BY dspace;
GRANT CONNECT TO dspace;
GRANT CREATE TABLE TO dspace;
GRANT CREATE SEQUENCE TO dspace;
GRANT CREATE VIEW TO dspace;
GRANT CREATE PROCEDURE TO dspace;
GRANT UNLIMITED TABLESPACE TO dspace;
```

Run DSpace Remotely OR Install DSpace on Virtual Machine

Once you have the (empty) database created on this Oracle Virtual Machine, you have two options (choose one):

1. You can run DSpace remotely (e.g. on your host machine), and just point it at this Oracle Virtual Machine (see: "Run DSpace Remotely")
2. OR, you can install DSpace directly on this Oracle Virtual Machine and run it there (see: "Install DSpace on Virtual Machine")

If you elect to run DSpace remotely, the following aliases may be helpful to you, if you are using a Unix or Linux-based OS for your host machine.

```
alias ostart='vboxmanage startvm "Oracle DB Developer VM" -type headless'
alias oshut='vboxmanage controlvm "Oracle DB Developer VM" savestate'
```

The *ostart* alias will start up your oracle database, the *oshut* alias will shut it down very quickly, essentially pausing your server in the state it is in.

If you are using Vagrant-DSpace as your development environment, you will want to create a new `vagrant.properties` file. Name it something else, `vagrant-oracle.properties` works. Here are the settings you'll want to use for the database.

```
db.name=oracle

# Uncomment the appropriate block below for your database.
# postgres
#db.driver=org.postgresql.Driver
#db.url=jdbc:postgresql://localhost:5432/dspace
#db.username=dspace
#db.password=dspace

# oracle
db.driver= oracle.jdbc.OracleDriver
db.url=jdbc:oracle:thin:@10.0.2.2:1521/orcl
db.username=username_you_chose_above
db.password=password_you_set_above
```

Things to note: you will want to comment out the Postgres configuration, and change the `db.name` to `oracle`. And you'll need to install the Oracle driver as per the usual.

The remainder of this how-to assumes that you wish to run a test DSpace instance in the virtual machine. If you prefer to run DSpace on the host machine instead, you could instead [set up port forwarding](#) for the Virtualbox appliance and use only the database in the virtual machine. In that case, skip the rest of this how-to.

Run DSpace Remotely (recommended)

If you chose this option, we are assuming you have DSpace installed elsewhere (e.g. on your host machine).

1. First setup port forwarding for this Oracle Virtual Machine
 - a. Open up your VirtualBox Manager
 - b. Select the Oracle Virtual Machine, and click the "Settings" button (at the top, looks like a gear)
 - c. Go to the "Network" settings, and click "Port Forwarding" button at the bottom
 - d. By default, you should already see a port forwarding rule for "sqlnet" which looks like this. You can modify that if you wish, or leave as-is.
 - i. sqlnet | TCP | Host Port: 1521 | Guest Port:1521
 - e. You also may wish to add a new rule for SSH access (port 22), something like this:
 - i. SSH | TCP | Host Port: 2200 | Guest Port: 22
 - f. Test out your SSH access. From your host machine type:
 - i. `ssh -p 2200 oracle@localhost (password: oracle)`
 - ii. Login as the "dspace" user: `sqlplus dspace/dspace`
2. On the machine where your DSpace is installed, you will need to do the following:
 - a. Download the latest Oracle OJDBC JAR from <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>
 - b. Install this JAR into your local Maven cache (~/.m2):

```
mvn install:install-file -Dfile=[path-to-downloads]/ojdbc6.jar -DgroupId=com.oracle -
DartifactId=ojdbc6 -Dversion=11.2.0.4.0 -Dpackaging=jar -DgeneratePom=true
# NOTE: If you are on Windows, each of the "-D" params need to be surrounded with double-quotes
```

- c. Update your `build.properties` database settings to refer to this Oracle database on the virtual machine. Pay close attention to the port, if you changed the default port forwarding rule for "sqlnet":

```
# oracle
db.driver=oracle.jdbc.OracleDriver
db.url=jdbc:oracle:thin:@//localhost:1521/orcl
db.username=dspace
db.password=dspace
```

d. Rebuild your DSpace with Oracle enabled:

```
mvn -Ddb.name=oracle package
# NOTE: If the version of the Oracle JDBC JAR has changed, you may need to update the Parent POM
to have the correct version
# See the "com.oracle" dependency: https://github.com/DSpace/DSpace/blob/master/pom.xml#L1082
# NOTE #2: If you are on Windows, the "-D" param needs to be surrounded with double-quotes
```

3. At this point, your DSpace should be "connected" with the Oracle Virtual Machine. You should be able to simply do a "fresh_install" of DSpace, and startup your Tomcat to see it in action

```
cd [src]/dspace/dspace-installer
ant fresh_install
```

4. You can also now login as the "dspace" database owner/user by simply SSHing into the Oracle Virtual Machine as follows:

```
ssh -p 2200 oracle@localhost (password: oracle)
sqlplus dspace/dspace
```

Install DSpace on Oracle Virtual Machine

Install build tools

The virtual machine image comes with ant preinstalled, but the version is too old (1.6.4). There is no maven installation on the machine.

Download the binary distributions of the ant and maven from <http://ant.apache.org/> and <http://maven.apache.org> and install them on the virtual machine, for example into ~/software. I symlinked the directories created by unpacking the tarballs so that I had ant in ~/software/ant and maven in ~/software/maven, so I'll use those paths in the rest of the instructions. You'll need to add ~/software/ant/bin and ~/software/maven/bin to the beginning of your PATH variable; eg put at the end of ~/.bash_profile:

```
export PATH=$HOME/software/ant/bin:$HOME/software/maven/bin:$PATH
```

You will need to log out and back in to make these settings apply, or type

```
source ~/.bash_profile
```

If ant -version shows that you are still running the old version of ant, try also adding this to ~/.bash_profile:

```
export ANT_HOME=$HOME/software/ant/bin
```

Build + install DSpace

Download the source distribution of DSpace (<https://github.com/DSpace/DSpace/archive/dspace-3.0.tar.gz> for DSpace 3.0) and unpack the tarball. I then renamed the directory to ~/dspace-src.

Create the directory ~/dspace.

Follow the standard [DSpace 3.x installation instructions](#) to configure, build and install DSpace. Change your build.properties file (leave everything else as-is):

```

dspace.install.dir=/home/oracle/dspace
...
db.name=oracle
...
# Uncomment the appropriate block below for your database.
# postgres
#db.driver=org.postgresql.Driver
#db.url=jdbc:postgresql://localhost:5432/dspace
#db.username=dspace
#db.password=dspace

# oracle
db.driver= oracle.jdbc.OracleDriver
db.url=jdbc:oracle:thin:@//localhost:1521/orcl
db.username=dspace
db.password=dspace
...

```

I also disabled the mail server in `~/dspace-src/dspace/config/dspace.cfg`.

It may not be readily apparent from the example config lines above, but the SID used by the Oracle VirtualBox image is "orcl".

Install Tomcat

Tomcat isn't installed on the virtual machine. Download the binary distribution from <http://tomcat.apache.org/>; I used Tomcat 6 and installed it into `~/software/tomcat` by unpacking the downloaded tarball and symlinking the directory. You will also need to copy the `ojdbc6.jar` file that you downloaded in the previous step into `~/software/tomcat/lib`.

Use your preferred method to link up Tomcat with the DSpace webapps that you need. I placed symlinks to `~/dspace/webapps/xmlui` and `~/dspace/webapps/solr` into `~/software/tomcat/webapps` after first removing everything from `~/software/tomcat/webapps`.

Start tomcat by running

```
cd ~/software/tomcat && ./bin/startup.sh
```

and your DSpace+Oracle instance should come up at <http://localhost:8080/xmlui/>.

Using a Docker Container

There is also a docker container image that you can pull from the Docker hub which will start up an Oracle database container that is configured to work with DSpace. In order to use this image, you need to [install Docker](#) on your development machine.

Starting the DSpace Oracle database container

Create a directory where the Oracle database server can store its data, for example `/tmp/dspace-oracle-data`. Then execute the following command to start the Oracle database server:

```
$ docker run -d -p 1521:1521 -v /tmp/dspace-oracle-data:/u01/app/oracle --name dspace-oracle atmire/dspace-oracle:11g2
```

This will create a new Docker container with name "dspace-oracle" that will run the Oracle database server.

Configure the DSpace database connection

Next, update your DSpace database configuration so that it has the following property values:

```

db.dialect = org.hibernate.dialect.Oracle10gDialect
db.schema = dspace
db.name = oracle
db.driver = oracle.jdbc.OracleDriver
db.url = jdbc:oracle:thin:@//localhost:1521/XE
db.username = dspace
db.password = dspace

```

Double check the database server is running and start Tomcat

Before starting Tomcat, you should make sure that the initialisation of the database server completed. You can check the logs of the database server by executing the command

```
$ docker logs dspace-oracle
```

You should see a message that starts with "Database ready to use.". If that is not the case, wait until the message appears (you need to re-execute the "logs" command each time).

When the database is ready, you can double check that your DSpace database configuration is correct by running this command and verifying the output:

```
$ bin/dspace database test

Attempting to connect to database using these configurations:
- URL: jdbc:oracle:thin:@//localhost:1521/XE
- Driver: oracle.jdbc.OracleDriver
- Username: dspace
- Password: [hidden]
- Schema: dspace

Testing connection...
Connected successfully!
```

If DSpace is configured correctly, you can then start Tomcat

Stop, restart or remove the database server.

Here are some commands to stop, restart or completely remove the Oracle database container:

To stop the container:

```
$ docker stop dspace-oracle
```

To start a previously stopped container:

```
$ docker start dspace-oracle
```

To completely remove the complete database container and its data (in order to start over with a clean database):

```
$ docker rm dspace-oracle
$ rm -rf /tmp/dspace-oracle-data
```