

OMERO.server installation on OS X with Homebrew

Overview

This walk-through demonstrates how to install OMERO on a clean Mac OS X system (10.9 or later) using Homebrew. Note that this demonstrates how to install OMERO.server *from the source code* via Homebrew, in addition to all its prerequisites.

These instructions are implemented in a series of [automated scripts](#) which install OMERO via Homebrew from a fresh configuration.

Prerequisites

Xcode

Homebrew requires the latest version of Xcode. Install **Xcode** and the Command Line Tools for Xcode from the App Store. If you have already installed it, make sure all the latest updates are installed.

Java

Oracle Java may be downloaded from the [Oracle website](#).

After installing JDK 7 or JDK 8, check your installation works by running:

```
$ java -version
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)

$ javac -version
javac 1.8.0_31
```

Requirements

All the requirements for OMERO will be installed under `/usr/local`. See also: Installation instructions on the [Homebrew wiki](#).

Install Homebrew and make sure `/usr/local/bin` is prepended to your **PATH**:

```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/ins
$ export PATH=/usr/local/bin:$PATH
```

Update Homebrew and run 'brew doctor' to fix potential issues beforehand:

```
$ brew update
$ brew doctor
```

Install git if not already present:

```
$ brew list | grep "\bgit\b" || brew install git
```

Install PostgreSQL database server:

```
$ export LANG=${LANG:-en_US.UTF-8}
$ export LANGUAGE=${LANGUAGE:-en_US:en}
$ brew install postgresql
```

You should install OMERO using Python 2.7 provided by Homebrew since it makes using Homebrew-provided modules simpler, for example the Ice python bindings needed by OMERO. For a more thorough description of the Homebrew solution, see the [Homebrew and Python](#) page. Note that the automated script linked above tests the OMERO installation using the Homebrew Python.

To install the Python provided by Homebrew:

Check that Python is working and is version 2.7:

```
$ which python
/usr/local/bin/python
$ python --version
Python 2.7.9
```

The installation of OMERO via Homebrew depends upon two alternate repositories containing extra formulae:

<https://github.com/Homebrew/homebrew-science> for the HDF5 formula and <https://github.com/ome/homebrew-alt> for all the OME-provided formulae and older versions of Ice. To add these, run:

```
$ brew tap homebrew/science
$ brew tap ome/alt
```

Note

The Homebrew formulae used below provide Python bindings. As described in [Homebrew and Python](#), you should **not** be in an active virtual environment when you `brew install` them.

See the [step01_deps.sh](#) script for the steps described above.

OMERO installation

OMERO 5.2.7 server

To install and deploy the 5.2.7 release of OMERO.server, run:

```
$ brew install omero52 --with-nginx --with-cpp
$ export PYTHONPATH=$(brew --prefix omero52)/lib/python
$ export ICE_CONFIG=$(brew --prefix omero52)/etc/ice.config
```

This will install the OMERO server to `/usr/local/Cellar/omero`, which means you will find the log files in `/usr/local/Cellar/omero/|release|/var/log`. The binaries will be linked to `/usr/local/bin`:

```
$ which omero
```

```
/usr/local/bin/omero
```

Install OMERO python dependencies:

```
$ pip install -r $(brew --prefix omero52)/share/web/requirements-py27-  
$ cd /usr/local  
$ bash bin/omero_python_deps
```

Start database server:

```
$ pg_ctl -D /usr/local/var/postgres -l /usr/local/var/postgres/server.
```

Create database and user:

```
$ createuser -w -D -R -S db_user  
$ createdb -E UTF8 -O db_user omero_database  
$ psql -h localhost -U db_user -l
```

Set database parameters in OMERO:

```
$ omero config set omero.db.name omero_database  
$ omero config set omero.db.user db_user  
$ omero config set omero.db.pass db_password
```

Create and run script to initialize the OMERO database:

```
$ export ROOT_PASSWORD=${ROOT_PASSWORD:-omero}  
$ export PSQL_SCRIPT_NAME=${PSQL_SCRIPT_NAME:-OMERO.sql}  
$ omero db script --password $ROOT_PASSWORD -f $PSQL_SCRIPT_NAME  
$ psql -h localhost -U db_user omero_database < $PSQL_SCRIPT_NAME  
$ rm $PSQL_SCRIPT_NAME
```

Set up OMERO data directory:

```
$ export OMERO_DATA_DIR=${OMERO_DATA_DIR:-~/OMERO.data}  
$ mkdir -p $OMERO_DATA_DIR  
$ omero config set omero.data.dir $OMERO_DATA_DIR
```

See the OMERO installation script [step02_omero.sh](#)

Development server

If you wish to build OMERO.server from source for development purposes, using the git repository, first use Homebrew to install the OMERO dependencies:

```
$ brew install --only-dependencies omero
```

The default version of Ice installed by the OMERO formula is currently Ice 3.5.

Prepare a place for your OMERO code to live, e.g.:

```
$ mkdir -p ~/code/projects
$ cd ~/code/projects
```

If you installed Ice 3.5, you will need to set **SLICEPATH** to be able to build the server, i.e. `export SLICEPATH=/usr/local/share/Ice-3.5/slice`.

If you want the development version of OMERO.server, you can clone the source code from the project's GitHub account to build locally:

```
$ git clone --recursive git://github.com/openmicroscopy/openmicroscopy
$ cd openmicroscopy && ./build.py
```

Note

If you have a GitHub account and you plan to develop code for OMERO, you should make a fork into your own account and then clone this fork to your local development machine, e.g.

```
$ git remote add git://github.com/YOURNAMEHERE/openmicroscopy
$ cd openmicroscopy && ./build.py
```

Then prepend the development `bin` directory to your **PATH** to pick the right executable:

```
$ export PATH=~/.code/projects/openmicroscopy/dist/bin:$PATH
```

and follow the steps for setting up the database and OMERO data directory as mentioned in the previous section.

OMERO.web

Basic setup for OMERO using Nginx:

```
$ export HTTPPORT=${HTTPPORT:-8080}
$ omero web config nginx-development --http $HTTPPORT > $(brew --prefi
```

See installation script [step03_nginx.sh](#)

For detailed instructions on how to deploy OMERO.web in a production environment such as Apache or Nginx please see [OMERO.web deployment](#).

Startup/Shutdown

If necessary start PostgreSQL database server:

```
$ pg_ctl -D /usr/local/var/postgres -l /usr/local/var/postgres/server.
```

Start OMERO:

Start OMERO.web:

```
$ omero web start
$ nginx -c $(brew --prefix omero52)/etc/nginx.conf
```

Now connect to your OMERO.server using OMERO.insight or OMERO.web

with the following credentials:

Stop OMERO.web:

```
$ nginx -c $(brew --prefix omero52)/etc/nginx.conf -s stop  
$ omero web stop
```

Stop OMERO:

See example script for a basic functionality test: [step04_test.sh](#)

Common issues

General considerations

If you run into problems with Homebrew, you can always run:

```
$ brew update  
$ brew doctor
```

Also, please check the Homebrew [Bug Fixing Checklist](#).

Below is a non-exhaustive list of errors/warnings specific to the OMERO installation. Some if not all of them could possibly be avoided by removing any previous OMERO installation artifacts from your system.

Database

Check to make sure the database has been created and 'UTF8' encoding is used

```
$ psql -h localhost -U db_user -l
```

This command should give similar output to the following:

List of databases

Name	Owner	Encoding	Collation	Ctype	Acc
omero_database	db_user	UTF8	en_GB.UTF-8	en_GB.UTF-8	
postgres	ome	UTF8	en_GB.UTF-8	en_GB.UTF-8	
template0	ome	UTF8	en_GB.UTF-8	en_GB.UTF-8	=c/ ome
template1	ome	UTF8	en_GB.UTF-8	en_GB.UTF-8	=c/ ome

(4 rows)

Macports/Fink

Warning: It appears you have MacPorts or Fink installed.

Follow uninstall instructions from the [Macports guide](#).

PostgreSQL

If you encounter this error during installation of PostgreSQL:

Error: You must ``brew link ossp-uuid' before postgresql can be instal

try:

```
$ brew cleanup
$ brew link ossp-uuid
```

szip

If you encounter an MD5 mismatch error similar to this:

```
==> Installing hdf5 dependency: szip
==> Downloading http://www.hdfgroup.org/ftp/lib-external/szip/2.1/src/
Already downloaded: /Library/Caches/Homebrew/szip-2.1.tar.gz
Error: MD5 mismatch
Expected: 902f831bcefb69c6b635374424acbead
Got: 0d6a55bb7787f9ff8b9d608f23ef5be0
Archive: /Library/Caches/Homebrew/szip-2.1.tar.gz
(To retry an incomplete download, remove the file above.)
```

then manually remove the archived version located under `/Library/Caches/Homebrew`, since the maintainer may have updated the file.

numexpr (and other Python packages)

If you encounter an issue related to numexpr complaining about NumPy having too low a version number, verify that you have not previously installed any Python packages using **pip**. In the case where **pip** has been installed before Homebrew, uninstall it:

```
$ sudo pip uninstall pip
```

and then try running `python_deps.sh` again. That should install **pip** via Homebrew and put the Python packages in correct locations.