Zero-Config Installation Notes (UNIX)

for Materials Data Curation System

These notes are for requirement 1.A7 of the Product Backlog (<http://goo.gl/ce9SNz>)

**For Operating System:**

UNIX

**Prerequisites**

Python 2.7.2 (brew install python)

pip ([http://pip.readthedocs.org/en/latest](http://pip.readthedocs.org/en/latest/))

Java 1.7 or higher

Install Python:

cd ~/

curl -L [http://www.python.org/ftp/python/2.7.2/Python-2.7.2.tgz > Python-2.7.2.tgz](http://www.python.org/ftp/python/2.7.2/Python-2.7.2.tgz)

tar xzf Python-2.7.2.tgz

cd python-2.7.2

./configure

make altinstall prefix=~/usr/local exec-prefix=~/usr/local

alias python='~/usr/local/bin/python2.7'

Install pip:

curl<https://bootstrap.pypa.io/get-pip.py> > get-pip.py

python get-pip.py

alias pip=~/usr/local/bin/pip

Any installed package via pip will now go under your '~/usr/local/lib/python2.7/site-packages' directory.

Install Java:

Make sure Java JDK 1.7 or higher is installed (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)

**Setup:**

We need pip to do the installation of the required dependencies. pip requires setuptools and it has to be installed first, before pip can run: <http://www.pip-installer.org/en/latest/installing.html>

Checkout MGI repository into ~/Develop/Workspaces

cd ~/Develop

pip install virtualenvwrapper (<http://virtualenvwrapper.readthedocs.org/en/latest/>)

export WORKON\_HOME=~/Develop/Envs

mkdir -p $WORKON\_HOME

source ~/usr/local/bin/virtualenvwrapper.sh

mkvirtualenv mgi

**For MongoDB (in a new terminal):**

cd ~/Develop/Workspaces/mgi/mdcs/

export WORKON\_HOME=~/Develop/Envs

source ~/usr/local/bin/virtualenvwrapper.sh

workon mgi

cd <PROJECT PATH>\mdcs

mkdir data

cd data

mkdir ts

mkdir db

Run MongoDB:

mongod --dbpath data/db/ --bind\_ip 127.0.0.1

To restrict access to mongodb database and ensure a better level of security, you may also:

* Add authentication/access control,
* Add traffic encryption,
* Have firewall rules.

You can find more about this topic: <http://docs.mongodb.org/manual/administration/security-checklist/>

**For Apache Jena (in a new terminal):**

cd ~/Develop/Workspaces/mgi/mdcs/

export WORKON\_HOME=~/Develop/Envs

source ~/usr/local/bin/virtualenvwrapper.sh

workon mgi

There are two options, depending on if you have the Java jar file or source code:

Option 1: Compile the source code and run the jar

cd <PROJECT PATH>/mdcs/rdf

javac -classpath "lib/\*" RdfServer.java SparqlServer.java JenaServers.java

java -cp ".:lib/\*" JenaServers -rdfserver\_endpoint 'tcp://127.0.0.1:5555' -sparqlserver\_endpoint 'tcp://127.0.0.1:5556' -tdb\_directory '/Users/ssy/Develop/Workspaces/mgi/mdcs/data/ts' -project\_uri 'http://www.example.com/'

Option 2: Run the jar

cd <PROJECT PATH>/mdcs/rdf

java -cp ".:lib/\*" JenaServers -rdfserver\_endpoint 'tcp://127.0.0.1:5555' -sparqlserver\_endpoint 'tcp://127.0.0.1:5556' -tdb\_directory '/Users/ssy/Develop/Workspaces/mgi/mdcs/data/ts' -project\_uri '<http://www.example.com/>'

If you get this error, then you need to upgrade your Java to at least version 1.7:

Exception in thread "main" java.lang.UnsupportedClassVersionError: JenaServers : Unsupported major.minor version 52.0

at java.lang.ClassLoader.defineClass1(Native Method)

at java.lang.ClassLoader.defineClass(ClassLoader.java:634)

at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)

at java.net.URLClassLoader.defineClass(URLClassLoader.java:277)

at java.net.URLClassLoader.access$000(URLClassLoader.java:73)

at java.net.URLClassLoader$1.run(URLClassLoader.java:212)

at java.security.AccessController.doPrivileged(Native Method)

at java.net.URLClassLoader.findClass(URLClassLoader.java:205)

at java.lang.ClassLoader.loadClass(ClassLoader.java:321)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:294)

at java.lang.ClassLoader.loadClass(ClassLoader.java:266)

Could not find the main class: JenaServers. Program will exit.

To query the RDF DB:

sh fuseki-server --loc=/Users/ssy/Develop/Workaces/mgi2/mgi/mdcs/data/ts /ts

**For Django (in a new terminal):**

cd ~/Develop/Workspaces/mgi/mdcs

export WORKON\_HOME=~/Develop/Envs

source ~/usr/local/bin/virtualenvwrapper.sh

workon mgi

pip install Django (<https://docs.djangoproject.com/en/1.6/topics/install/>)

pip install mongoengine (<http://docs.mongoengine.org/guide/installing.html>)

pip install django-auth-ldap (<http://pythonhosted.org/django-auth-ldap/>)

pip install django-dajax (<http://django-dajax.readthedocs.org/en/latest/installation.html>)

pip install xlrd (needed for Microsoft Excel to HDF5 translation/mapping)

pip install xmltodict (<https://github.com/martinblech/xmltodict>)

pip install pyzmq (<https://github.com/zeromq/pyzmq>)

pip install python-dateutil (<https://pypi.python.org/pypi/python-dateutil>)

pip install lxml (<http://lxml.de/>)

If you get the error “clang error: linker command failed”, then run the following command instead:

STATIC\_DEPS=true pip install lxml (<http://lxml.de/installation.html>)

pip install djangorestframework (<http://www.django-rest-framework.org/>)

pip install django-rest-framework-mongoengine (<https://pypi.python.org/pypi/django-rest-framework-mongoengine>)

pip install django-rest-swagger (<https://github.com/marcgibbons/django-rest-swagger>)

python manage.py syncdb

Answer yes to:

You just installed Django's auth system, which means you don't have any superusers defined.

Would you like to create one now? (yes/no):yes

Run software:

python manage.py runserver

**Access:**

For Materials Data Curation System, Go to: <http://127.0.0.1:8000/>

For Materials Data Curation Administration, Go to: <http://127.0.0.1:8000/admin/>