

# Installing DSpace on FreeBSD

- 1 Before we start
- 2 Required software
- 3 Configuration rollercoaster
  - 3.1 Configure system
  - 3.2 Configure PostgreSQL
  - 3.3 Configure JAVA
  - 3.4 Configure Tomcat
  - 3.5 Configure Apache
- 4 Install Dspace
- 5 Handle
- 6 Clean up and daemons startup
- 7 Final notes

## 1 Before we start

- Tomcat runs as "www" by default and i found a bit easier to run dspace under "www" user too. If You decided to create "dspace" user then pay attention to filesystem permissions. Using "www" may not be good idea when server is accessible by any other user(s) than system administrator (s).
- If You like to run dspace on port 80/443, you'll need Apache web server. Tomcat itself uses on ports somewhere 8000 ... 9000.
- Since dspace does not come from ports, there's no exact or good place for it. Think something out. Solaris often uses "/opt" or "/export". You may create those mountpoints. I thought that "/data" is fine to put most of dspace related stuff there. You can create mountpoint "/dspace" but i don't think it's a good idea to put dspace software directly to mountpoint. It's better to have dspace in some kind sub-directory (eg /mntpnt/dspace ). Also avoid construction like /dspace/dspace, which is just confusing.
- I decided to create slices like that:

```
tcsh# df -h -t ufs
Filesystem  Size  Used  Avail Capacity  Mounted on
/dev/da0s1a 1.9G  489M  1.3G   27%  /
/dev/da0s1d 496M   84K  456M    0%  /tmp
/dev/da0s1e 1.4G  826M  537M   61%  /usr/local
/dev/da0s1f 1.9G  160M  1.6G    9%  /var

tcsh# df -h -t zfs
Filesystem  Size  Used  Avail Capacity  Mounted on
data        63G  2.1G  60G    3%  /data
```

/usr/src and /usr/ports are mounted from NFS server if needed. If You need those too, then change partition sizes accordingly and maybe use separate mountpoint for "/usr" instead of "/usr/local". "/data" is second disk. However - layout above should give picture how much required software uses disk space.

- **NB! Be extremely careful if using copy-paste!** Also note that wiki page may eat some specific chars.

## 2 Required software

Install them in that order. You can find them from /usr/ports.

```
lang/perl5.14
lang/python27
www/apache22
databases/postgresql91-server
[ databases/postgresql91-contrib <- optional, but may become handy ]
java/openjdk6
www/tomcat7
devel/apache-ant
www/mod_jk
devel/maven3
shells/bash
```

## 3 Configuration rollercoaster

### 3.1 Configure system

```
tcsh# echo "fdesc /dev/fd fdescs rw 0 0" >> /etc/fstab
```

```
tcsh# zpool create data /dev/da1
```

Since we use "www" user to run dspace, we need to provide working shell and home for it. But let's also tighten access.

```
tssh# grep AllowGroups /etc/ssh/sshd_config
AllowGroups wheel
```

```
tssh# mkdir /var/log/apache2
tssh# chown www /var/log/apache2
tssh# mkdir -p /data/home/www
tssh# pw usermod www -d /data/home/www
tssh# echo "exit" > /data/home/www/.login
tssh# chsh -s /bin/sh www
```

Several scripts from dspace are using "/bin/bash". To make them happy:

```
tssh# ln -s /usr/local/bin/bash /bin/bash
```

### 3.2 Configure PostgreSQL

```
tssh# echo 'postgresql_enable="YES"' >> /etc/rc.conf
tssh# echo 'postgresql_data="/data/pgsql"' >> /etc/rc.conf
tssh# mkdir /data/pgsql
tssh# chown -R postgres:postgres /data/pgsql/
tssh# /usr/local/etc/rc.d/postgresql initdb
```

By default PostgreSQL uses "local0", but if You are using pf firewall logging also, then change to "local3" for example to avoid messing up logs.

```
tssh# egrep "listen|syslog_facility" /data/pgsql/postgresql.conf
listen_addresses = 'localhost'
syslog_facility = 'LOCAL3'
```

```
tssh# egrep -v "#|^$" /data/pgsql/pg_hba.conf
local all          postgresql          trust
host dspace db     dspace      127.0.0.1/32    md5
```

Start database engine and create database user "dspace":

```
tssh# /usr/local/etc/rc.d/postgresql start
```

```
tssh# createuser -U postgres -S -d -R -P dspace
#-S = --no-superuser
#-d = user will be allowed to create databases
#-R = --no-createrole
#-P createuser will issue a prompt for the password
```

or

```
tssh# createuser -U postgres -P dspace
Enter password for new role: s0mepw
Enter it again: s0mepw
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) y
Shall the new role be allowed to create more new roles? (y/n) n
```

Create database called "dspace db":

```
tssh# createdb -U postgres -O dspace -E UNICODE dspace db
```

### 3.3 Configure JAVA

```
tssh# echo "JAVA_HOME=/usr/local/openjdk6/" >> /usr/local/etc/javavm_opts.conf
```

Following affects only commands from shell. **Make sure that both parameters (-Xmx and -Xms) do have same value and at least 512m!** If they differ and/or are less than 512, then You may encounter problems later on if using "ant update" (if upgrading dspace).

```
tssh# echo 'JAVA_OPTS="-Xmx512m -Xms512m"' >> /usr/local/etc/javavm_opts.conf
```

### 3.4 Configure Tomcat

Open file `/usr/local/apache-tomcat-7.0/conf/server.xml` with Your favorite vi. **NB! Pay attention to UTF!** Locate relevant lines and update to be:

```
<Connector port="8080" protocol="HTTP/1.1"
  maxThreads="150"
  minSpareThreads="25"
  maxSpareThreads="75"
  enableLookups="false"
  redirectPort="8443"
  acceptCount="100"
  connectionTimeout="20000"
  disableUploadTimeout="true"
  URIEncoding="UTF-8" />
```

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8" />
```

Once again i found more reasonable not to copy (or symlink) webapps to tomcat appBase dir as suggests dspace official documentation. Instead i'm changin tomcat appBase to point to dspace webapps. Also put tomcat logs with other www/apache logs. Original lines are commented out and my lines marked bold.

```
<!-- <Host name="localhost" appBase="webapps" -->
<Host name="localhost" appBase="/data/dspace/webapps"
  unpackWARs="true" autoDeploy="true">
  <!-- <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
    prefix="localhost_access_log." suffix=".txt"
    pattern="%h %l %u %t "%r" %s %b" /> -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="/var/log/apache2"
    prefix="tomcat_access." suffix=".log"
    pattern="%h %l %u %t "%r" %s %b" />
```

Finally configure worker:

```
tcsh# grep -v "#" /usr/local/etc/apache22/workers.properties
worker.list=localhost-worker
worker.localhost-worker.port=8009
worker.localhost-worker.host=localhost
worker.localhost-worker.type=ajp13
worker.localhost-worker.lbfactor=1
```

### 3.5 Configure Apache

Some lines are omitted from output. Also configure "apache22/extra/httpd-mpm.conf" and "httpd-default.conf" to suit You. Also don't forget apache certificates.

```

tcsh# egrep -v "#|^$" /usr/local/etc/apache2/httpd.conf
/*
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
LoadModule jk_module      libexec/apache2/mod_jk.so
/*
ErrorLog "/var/log/apache2/httpd-error.log"
/*
CustomLog "|/usr/local/sbin/rotatelog -l /var/log/apache2/httpd-access_%Y-%m-%d.log 86400" combined
/*
Include etc/apache2/extra/httpd-mpm.conf
Include etc/apache2/extra/httpd-default.conf
Include etc/apache2/extra/httpd-ssl.conf
/*
Include etc/apache2/Includes/*.conf
NameVirtualHost *:80
<IfModule jk_module>
    # relative path to /usr/local
    JkWorkersFile etc/apache2/workers.properties
    JkShmFile /var/run/jk-runtime-status
    JkLogLevel error
    JkLogFile /var/log/apache2/mod_jk.log
</IfModule>
<VirtualHost *:80>
ServerName dspace.example.com
    DocumentRoot /usr/local/www/apache2/data
    <IfModule jk_module>
        JkMount /xmlui localhost-worker
        JkMount /xmlui/* localhost-worker
        JkMount /solr localhost-worker
        JkMount /solr/* localhost-worker
        JkMount /oai localhost-worker
        JkMount /oai/* localhost-worker
    </IfModule>
    RewriteEngine On
    RewriteRule ^/$ /xmlui/ [PT]
    RewriteRule ^/$ /solr/ [PT]
    RewriteRule ^/$ /oai/ [PT]
    RewriteCond http://%{HTTP_HOST}%{REQUEST_URI} (.*)-login(.*) [OR]
    RewriteCond http://%{HTTP_HOST}%{REQUEST_URI} (.*)/register(.*) [OR]
    RewriteCond http://%{HTTP_HOST}%{REQUEST_URI} (.*)/forgot(.*)
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
    #
    CustomLog "|/usr/local/sbin/rotatelog -l /var/log/apache2/dspace.example.com-access-%Y-%m-%d.log 86400" combined
    ErrorLog /var/log/apache2/dspace.example.com-error.log
</VirtualHost>

```

```

tcsh# egrep -v "#|^$" /usr/local/etc/apache2/extra/httpd-ssl.conf
Listen 443
/*
<VirtualHost _default_:443>
ServerName dspace.example.com:443
ServerAdmin hostmaster@example.com
DocumentRoot "/usr/local/www/apache2/data"
<IfModule jk_module>
    JkMount /xmlui localhost-worker
    JkMount /xmlui/* localhost-worker
    JkMount /solr localhost-worker
    JkMount /solr/* localhost-worker
    JkMount /oai localhost-worker
    JkMount /oai/* localhost-worker
    RewriteEngine On
    RewriteRule ^/$ /xmlui/ [PT]
    RewriteRule ^/$ /solr/ [PT]
    RewriteRule ^/$ /oai/ [PT]
</IfModule>
ErrorLog "|/usr/local/sbin/rotatelog /var/log/apache2/https-error-%Y-%m-%d.log 5M"
TransferLog "|/usr/local/sbin/rotatelog /var/log/apache2/https-access-%Y-%m-%d.log 86400"
/*
SSLCertificateFile "/usr/local/etc/apache2/certs/dspace.example.com.crt"
SSLCertificateKeyFile "/usr/local/etc/apache2/certs/dspace.example.com.key"
SSLCertificateChainFile "/usr/local/etc/apache2/certs/dspace-bundle.example.com.crt"
/*
<Directory "/data/dspace/webapps/xmlui">
    SSLOptions +StdEnvVars +ExportCertData
</Directory>

```

## 4 Install Dspace

```
tssh# mkdir /data/dspace
tssh# cd /data
tssh# fetch -o dspace-1.8.1-src-release.tar.gz *http://sourceforge.net/projects/dspace/files/Dspace%20Stable/1.8.1/dspace-1.8.1-src-release.tar.gz/download* (http://sourceforge.net/projects/dspace/files/Dspace%20Stable/1.8.1/dspace-1.8.1-src-release.tar.gz/download*)
tssh# tar xzf dspace-1.8.1-src-release.tar.gz
tssh# cd /data/dspace-1.8.1-src-release/dspace/config/
```

Configure whatever authentication You need. In following example dspace is configured to use LDAP.

```
tssh# egrep -v "#|^$" modules/authentication.cfg
plugin.sequence.org.dspace.authenticate.AuthenticationMethod = org.dspace.authenticate.LDAPHierarchicalAuthentication
```

Open /data/dspace-1.8.1-src-release/dspace/config/dspace.cfg and make Your changes:

```
dspace.dir = /data/dspace
dspace.hostname = dspace.example.com
dspace.baseUrl = http://dspace.example.com
dspace.url = ${dspace.baseUrl}/xmlui
dspace.name = Dspace at Example.Com
db.name = postgres
db.url = jdbc:postgresql://localhost:5432/dspacedb
db.driver = org.postgresql.Driver
db.username = dspace
db.password = s0mepw
db.maxconnections = 30
db.maxwait = 5000
db.maxidle = -1
db.statementpool = true
mail.server = smtp.example.com
mail.server.port = 25
mail.from.address = dspace-noreply@example.com
feedback.recipient = dspace-help@example.com
mail.admin = dspace-help@example.com
alert.recipient = postmaster@example.com
registration.notify = dspace-help@example.com
mail.charset = UTF-8
mail.allowed.referrers = localhost,dspace.example.com
mail.server.disabled = false
default.language = en_US
assetstore.dir = ${dspace.dir}/assetstore
log.init.config = ${dspace.dir}/config/log4j.properties
log.dir = /var/log/apache2/
search.dir = ${dspace.dir}/search
/*
handle.canonical.prefix = http://hdl.handle.net/
handle.prefix = 12345
handle.dir = ${dspace.dir}/handle-server
/*
upload.max = 536870912
default.locale = en
xmlui.supported.locales = en
xmlui.force.ssl = true
xmlui.user.registration=false
```

Configure LDAP module. As i'm writing its not possible to configure multiple ldap servers in order to achieve failover (eg. ldap://ldapserver1 ldapserver2/?blah?blah).

```
tssh# egrep -v "#|^$" modules/authentication-ldap.cfg
enable = true
autoregister = true
provider_url = ldaps://myldap.example.com/
id_field = uid
object_context = ou=people,dc=example,dc=com
search_context = ou=people,dc=example,dc=com
email_field = mail
surname_field = sn
givenname_field = givenName
phone_field = telephoneNumber
search_scope = 2
search.user = cn=ldap-bind,cn=Users,dc=example,dc=com
search.password = s0mepw2
netid_email_domain = @example.com
```

As You can see, i'm using LDAPS. We'll be back to it later on.

Following command fetches software from internet in order to build dspace. This soft will be placed under \$HOME/.m2/ directory. In my case /root/.m2/. If You want to, You can build dspace as "www" user. I'm doing it as root.

```
tssh# /data/dspace-1.8.1-src-release
tssh# mvn package
tssh# cd /data/dspace-1.8.1-src-release/dspace/target/dspace-1.8.1-build/
tssh# ant fresh_install
```

As looking from my notes there was an issue with creating PostgreSQL database (PL/pgSQL related). Seems that following helped out. However - i can't verify or confirm it at the moment.

```
tssh# dropdb -U postgres dspace
tssh# createdb -U postgres -O dspace -E UNICODE dspace
tssh# psql -h localhost -U dspace -f /data/dspace-1.8.1-src-release/dspace/etc/postgres/database_schema.sql dspace
```

And finally set proper permissions:

```
tssh# chown -R www:www /data/dspace
```

Just in case verify /data/dspace/config/log4j.properties doesn't bug You. Remove unneeded "/" There may be 3 erratic lines like this one:

```
/var/log/apache2/cocoon.log
```

Also verify that /data/dspace/config/modules/authentication.cfg and authentication-ldap.cfg are correct.

Since i like to keep all dspace related things in one place and i have pretty small /usr/local:

```
tssh# mkdir -p /data/dspace/tc-webinf/work/upload-dir
tssh# mkdir -p /data/dspace/tc-webinf/work/cache-dir
tssh# chown -R www:www /data/dspace/tc-webinf

tssh# grep dspace /data/dspace/webapps/xmlui/WEB-INF/cocoon/properties/core.properties
org.apache.cocoon.uploads.directory=/data/dspace/tc-webinf/work/upload-dir
org.apache.cocoon.cache.directory=/data/dspace/tc-webinf/work/cache-dir
org.apache.cocoon.work.directory=/data/dspace/tc-webinf/work/
```

Don't forget this one if You upgraded Your dspace - cocoon may fill /usr/local.

If needed, configure OAI also: /data/dspace/config/oaicat.properties:

```
/*
Crosswalks.mods=org.dspace.app.oai.PluginCrosswalk
Crosswalks.mets=org.dspace.app.oai.PluginCrosswalk
Crosswalks.qdc=org.dspace.app.oai.PluginCrosswalk
```

Set up crontabs. PATH is required.

```
dspace ~ # crontab -l -u www
#dspace www crontab
PATH=/data/dspace/bin:/bin:/usr/bin:/usr/local/bin:/sbin:/usr/sbin:/usr/local/sbin
#Send out subscription e-mails at 01:00 every day
0 1 * * * /data/dspace/bin/dspace sub-daily
#Run the media filter at 02:00 every day
0 2 * * * /data/dspace/bin/dspace filter-media > /dev/null 2>&1
#Run the checksum checker at 03:00
0 3 * * * /data/dspace/bin/dspace checker -lp > /dev/null 2>&1
#Mail the results to the sysadmin at 04:00
0 4 * * * /data/dspace/bin/dspace checker-emailer -c > /dev/null 2>&1
#Run stat analysis
0 1 * * * /data/dspace/bin/dspace stat-general
0 1 * * * /data/dspace/bin/dspace stat-monthly
0 2 * * * /data/dspace/bin/dspace stat-report-general
0 2 * * * /data/dspace/bin/dspace stat-report-monthly
```

Now install certificates required to use LDAPS. Make sure that You have JAVA\_HOME set:

```

tcsh# set JAVA_HOME=/usr/local/openjdk6
tcsh# echo $JAVA_HOME
/usr/local/openjdk6
tcsh# keytool -import -file /tmp/myldap-clients.example.com.crt -alias myldap.example.com -keystore $JAVA_HOME/jre/lib/security/cacerts
Enter keystore password: 'changeit' <- by default without '-es!'
/*
Trust this certificate? [no]: yes
Certificate was added to keystore
tcsh# keytool -list -keystore $JAVA_HOME/jre/lib/security/cacerts
tcsh# rm -f /tmp/myldap-clients.example.com.crt

```

## 5 Handle

If You are using "handle" also, then:

```
tcsh# /data/dspace/bin/dspace make-handle-config /data/dspace/handle-server
```

Create `/usr/local/etc/rc.d/handle` with following content. This script runs handle service as "www" user.

```

#!/bin/sh
#
# PROVIDE: handle
# REQUIRE: NETWORKING tomcat7
# KEYWORD: shutdown
#
# handle_server_enable="YES"
#

./etc/rc.subr

name="handle_server"
start_cmd="${name}_start"
stop_cmd="${name}_stop"
rcvar=`set_rcvar`

command="/data/dspace/bin/start-handle-server"

handle_server_start()
{
    if [ -x ${command} ]; then
        pid=`ps -axuwww | grep -v grep | grep handle-server | awk '{ print $2 }'`
        if [ "${pid}"X = "X" ]; then
            su - www -c ${command}
        else
            echo "Handle server is already running."
        fi
    fi
}

handle_server_stop()
{
    pid=`ps -axuwww | grep -v grep | grep handle-server | awk '{ print $2 }'`
    if [ "${pid}"X != "X" ]; then
        pid_owner=`ps -axu | grep -v grep | grep -w $pid |awk '{ print $1 }'`
        if [ "${pid_owner}" = "www" ]; then
            kill -15 ${pid}
            sleep 1
        fi
    else
        echo "Handle server is not running?"
    fi
}

# set defaults

handle_server_enable=${handle_server_enable:-"NO"}
load_rc_config "${name}"

run_rc_command "$1"

```

## 6 Clean up and daemons startup

```

tcsh# cd /data/dspace-1.8.1-src-release
tcsh# mvn clean
tcsh# rm -r /root/.m2

```

Enable all required services at startup - `/etc/rc.conf`. Once again pay attention to **UTF** and **make sure that "-Xmx" and "-Xms" are at least 512M and both do have same values!**

```
apache22_enable="YES"
tomcat7_enable="YES"
tomcat7_java_opts="-Xmx512M -Xms512M -XX:MaxPermSize=128M -Dfile.encoding=UTF-8"
tomcat7_catalina_log=">> /var/log/apache2/catalina- date +%Y-%m-%d'.log 2>&1"
tomcat7_catalina_tmpdir="/tmp"
handle_server_enable="YES"
postgresql_enable="YES"
postgresql_data="/data/pgsql"
```

```
tcsh# sync; sync; reboot
```

## 7 Final notes

- If You should later on upgrade "openjdk", then You need to import LDAP certificate again - you'll lose it!
- If You should upgrade mod\_jk port, then dont forget to uncomment "#LoadModule jk\_module.... " line!
- After dspace upgrade dont forget cocoon: /data/dspace/webapps/xmlui/WEB-INF/cocoon/properties/core.properties
- Implement backups and monitoring!
- Implement firewall. If using pf:

```
WEB_PORTS="{ 80, 443 }"
HANDLE_PORTS="{ 2641, 8000 }"
# www
pass in log quick on $EXT_IF proto tcp from any to port $WEB_PORTS
# dspace handle service
pass in log quick on $EXT_IF proto tcp from any to port $HANDLE_PORTS
```

Maybe You need SSH too, but in general keep blocking.

- Please read carefully [dspace documentation](#).