

Installing DSpace on SLES9

<?xml version="1.0" encoding="utf-8"?>

<html>

<p>This is a quick-guide for installing Dspace 1.3.2/1.4.1 on a SuSE Linux Enterprise 9 (SLES9) server with Tomcat and PostgreSQL.

It also includes comments to the installation guide for the Statistics add-on [and the Request Copy add-on <http://wiki.dspace.org/index.php/RequestCopy>] from Universidade of Minho</p>

<h2>Prerequisites</h2>

Java 1.4 or later(standard SDK is OK)

Apache Ant 1.6.2 or later

PostgreSQL 8.x

Jakarta Tomcat 4.x/5.x(other app.servers are supported)

<h2>PostgreSQL 8.1.4</h2>

PostgreSQL might be pre-installed.

To compile it from scratch, do as follows:

Check the requirements - <http://www.postgresql.org/docs/8.1/interactive/install-requirements.html> and install missing packages, if any. (Normally only 'readline-devel' is missing)

If you are upgrading, you'll definitely need to read <http://www.postgresql.org/docs/8.1/interactive/install-upgrading.html>

Create a user to run the database server as - typically named *postgres* with home directory */usr/local/pgsql*. Add the following to this users .bashrc file if you are going to install in the default location:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/pgsql/lib
export PATH=$PATH:/usr/local/pgsql/bin
export MANPATH=$MANPATH:/usr/local/pgsql/man
export POSTGRES_HOME=/usr/local/pgsql
```

While you are creating users, do also create the Dspace user - called *dspace* with home directory */home/dspace*. Create a group for this user as well.

To this users .bashrc or equivalent, add:

```
export PATH=$PATH:/usr/local/pgsql/bin
```

Download the tar-ball and extract it to a location of your own choice.

Configure it by running:

```
./configure
```

If everything is OK, compile it by running:

```
gmake
```

Check the new build. This cannot be done as root. Use an unprivileged user(for example the postgres user you created earlier)

```
gmake check
```

As root, install the database server by running:

```
gmake install
```

Create the directory where you are going to store the database and make the postgres user owner

```
mkdir /dspace/db; chown postgres:postgres /dspace/db
```

As the postgres user create a new database cluster

```
/usr/local/pgsql/bin/initdb -D /dspace/db
```

Uncomment the line starting with "listen_addresses = 'localhost'" in /dSPACE/db/postgresql.conf.

In /dSPACE/db/pg_hba.conf, add the line

```
"host dSPACE dSPACE 127.0.0.1 255.255.255.255 md5"
```

You'll need to set kernel parameters to tune the installation. The following are minimum on Linux. See the documentation (<http://www.postgresql.org/docs/8.1/interactive/kernel-resources.html>)

There are several ways of doing this, one option is to add the parameters to /etc/sysctl.conf (a reboot is needed):

```
kernel.shmmax=134217728
kernel.shmall=2097152
vm.overcommit_memory=2
```

You can check the settings by running `sysctl -A | grep kernel | sort | less` whilst logged in as root.

As the postgres user, startup the database instance by running (create a start/stop script and link to a run level)

```
/usr/local/pgsql/bin/postmaster -D /dSPACE/db >logfile 2>&1 &
```

If no error messages are shown, you are ready for the next step!

<h2>Tomcat 5.x</h2>

As with PostgreSQL, Tomcat may already be installed on your server. In any case, Tomcat need to run as the same user as DSPACE, and not as the default Tomcat-user.

So you will either need to install a new version of Tomcat as the user dSPACE, or change ownership on all directories that already are owned by tomcat (chown -R dSPACE:dSPACE /usr/share/tomcat). There are quite a few symbolic links in /usr/share/tomcat, so you'll have to change ownership on them as well.

Edit the /usr/share/tomcat/conf/server.xml file so that searching and browsing of multi-byte UTF-8 are done correctly:

The <connector> element for connection on port 8080 should read:

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

(The string 'URIEncoding="UTF-8"' is added)

There are several ways of starting Tomcat, in one of the startup-scripts, make sure that the following is set:

```
JAVA_OPTS="-Xmx512M -Xms64M -Dfile.encoding=UTF-8"
```

You can set this by for example adding the following line to /usr/share/tomcat/bin/startup.sh

```
export JAVA_OPTS="-Xmx512M -Xms64M -Dfile.encoding=UTF-8"
```

Our DSPACE installation is set up to run on Apache and Tomcat with the mod_jk connector.

To do this, read <http://wiki.dspace.org/ModJk>

We recommend not to compile and install the required stuff from the source. Use the rpm's from SuSE.

The config files are located a bit differently in SLES:

```
/etc/tomcat/base/workers.properties
```

```
/etc/apache2/conf.d/jk.conf
```

```
/usr/lib/apache2/mod_jk.so
```

If you don't want the last bit of the URL(/dSPACE), you'll need to configure the DSPACE application as Tomcat's ROOT application.

To do this, you'll need to rename the two dSPACE*.war files you soon will be copying to /usr/share/tomcat/webapps/ to ROOT*.war, and make a copy of the existing server.xml in /usr/share/tomcat/conf and call it ROOT.xml

Start Tomcat (as the dSPACE user) by running `/usr/share/tomcat/bin/startup.sh` or use SLES9 standard rc-script as root. Make sure Tomcat starts as the dSPACE user. Link script(s) to your preferred runlevel

<h2>DSpace 1.3.2</h2>

If you don't have a local CVS repository, just skip the CVS related stuff...

We are also using a custom deploy script so we easily can deploy the application to either the live or the test server. Follow the standard DSPACE installation guide for the steps after the *ant fresh_install* if you chose not to use such a script.

As the dspace user:

Download the tarball and extract it to a location of your choice.

Change directory to where you extracted the tarball and import the code into CVS by doing

```
cvs import -I ! -m "Import DSpace 1.3.2" dspace-1_3_2 SourceForge dspace-1_3_2
```

Go to ~dspace/cvs and checkout the source by doing

```
cvs checkout dspace-1_3_2 dspace-1_3_2
```

Also download the PostgreSQL JDBC driver for the version of PostgreSQL you are running and for JDBC2 (<http://jdbc.postgresql.org/download.html>)

Copy the driver into the lib directory of the dspace source you just checked out and add it to CVS.

Create a dspace database, owned by the dspace PostgreSQL user(do this as postgres):

```
createuser -U postgres -d -A -P dspace ; createdb -U dspace -E UNICODE dspace
```

(You'll need to type a password for the database, do not choose the same as the dspace unix user password...)

This next bit is specific to our setup and is done to be able to use our custom deploy script.

*Copy the default ~dspace/cvs/dspace-1_3_2/config/dspace.cfg to ~dspace/cvs/dspace-1_3_2/config/dspace.cfg.LIVESERVER
Also make a copy of the same file in the same location and name it dspace.cfg.TESTSERVER. You will need this one when you are deploying to the test server. It will be different to the one for the liveserver.*

Edit dspace.cfg.LIVESERVER (and dspace.cfg.TESTSERVER) - or just the normal dspace.cfg file in most cases

You will at least need to modify the following entries:

```
dspace.url  
dspace.hostname  
dspace.name  
db.password (the password you entered in the previous step)  
mail.server  
mail.from.address  
feedback.recipient  
mail.admin  
alert.recipient (not essential but very useful!)
```

Edit ~dspace/cvs/config/dstat.cfg to read the correct dspace.log directory.

Create a directory for the installation and change ownership to the dspace user and group.

```
mkdir /dspace; chown dspace:dspace /dspace
```

As the dspace user, compile and install DSpace:

```
cd ~dspace/cvs/dspace-1_3_2 ; ant fresh_install
```

The most likely thing to go wrong here, is the database connection.

You will also need to edit the ~dspace/cvs/dspace-1_3_2/bin/dsrun file to include full path to java.

Because we have a custom script that will actually take care of the deployment, it's now time to run this.... Go to ~dspace and run

```
./deploy_dspace live 1_3_2
```

This will copy files across, restart tomcat, etc.

Create an initial administrator account:

```
/dspace/bin/create-administrator
```

Dspace should be running happy on (depending on your configuration):

<http://servername.domain:8080/dspace>

<http://servername.domain/dspace>

or

<http://servername.domain:8080>

<http://servername.doamin>

You can access the admin GUI by adding /dspace-admin to one of the above URL's

<h2>Handle Server</h2>

A production DSpace repository must include a handle server. Follow the DSpace installation guide when setting this up.

NB! DO NOT encrypt the passwords when asked in the setup procedure.

The handle server will need TCP port 2641 and 8000 opened in the firewall.

The handle.jar file that comes with Dspace is old, download the new one as described in the Dspace doc's, copy it to ~dspace/cvs/dspace-1_3_2/lib and add it to CVS as you did with the JDBC driver.

<h2>Cronjobs</h2>

Set up cronjobs as described in the DSpace Installation Guide

<h2>Deploying Dspace</h2>

Normally you would manually run 'ant update', copy the *.war files to tomcat and then restart tomcat when a redeployment of the application is required. It's quite a lot of stuff to remember, so we have a script that will do that for you.

The code can be downloaded here http://wiki.dspace.org/static_files/a/aa/Deploy_dspace.

<h2>Statistics Add-on</h2>

The statistics add-on is developed by the University of Minho in Portugal. It can be downloaded from <http://wiki.dspace.org/index.php/StatisticsAddOn>. The install instructions are included in the tar-ball. Remember to add new files/check changes into CVS.

Some comments to the install instructions:

1. When installing PL/Java in Step 1, you'll also need to install the plpgsql language (as the postgres user) by doing

```
createlang -U postgres plpgsql dspace
```

2. To deploy PL/Java, use the deployer program and not the sql script provided. As the postgres user run:

```
java -cp </path_to>/postgresql.jar:/usr/share/pljava/deploy.jar org.postgresql.pljava.deploy.Deployer -install -database dspace
```

3. In Step 2, the line psql -U postgres -d dspace should be

```
psql dspace (Logged in as the postgres user).
```

The line psql -u dspace should be

```
psql dspace (Logged in as the dspace user)
```

4. In Step 3, make sure you have a copy of the files in the [dspace_source](#)/src directory before you copy the edited files from the add-on source.

I would recommend to change them by hand as the code seem to be an old version. Look in the files provided by Minho and cut and paste in the code regarding the add-on. It's comments in the files, so it should be fairly easy to find. Diff may also help you.

5. The path to where you will need to paste in the <tag> code from the install guide is wrong:

```
It should be $DSPACE_SRC/jsp/WEB-INF/dspace-tags.tld
```

6. In Step 5, apply the changes by running the deploy script in ~dspace/ as it will take care of the copying of files, etc.

7. The database function 'stats_is_institution' is calling a build in function named 'instr'. The trouble is that 'instr' doesn't exist.

Run the sql found here http://wiki.dspace.org/static_files/d/d5/Instr.sql to create a workaround.

<h2>RequestCopy Add-On</h2>

This add-on makes it possible for repository administrators to restrict access to a certain bitstream(an embargoed paper) - but a user that really wants a copy of it can request it from the author via e-mail. It's developed by the University of Minho, and can be downloaded from <http://wiki.dspace.org/index.php/RequestCopy>

The install instructions are included in the tar-ball. Remember to add new files/check changes into CVS

Some comments to the install instructions:

1. In Step 1, be careful when copying the src directory as it will overwrite two files in the DSpace sources. I recommend to change \$DSPACE_SRC/src/org/dspace/app/webui/servlet/BitstreamServlet.java and \$DSPACE_SRC/src/org/dspace/app/webui/jsptagItemTag.java by hand do make sure you don't overwrite any Dspace stuff. Look in the files provided by Minho and cut and paste in the code regarding the add-on. It's comments in the files, so it should be fairly easy to find. Diff may also help you.

2. In Step 2, you only need to copy the servlet tags to \$DSPACE_SRC/etc/dspace-web.xml as the other file they mention doesn't exist.

3. In Step 5, apply the changes by running the deploy script in ~dspace/ as it will take care of the copying of files, etc.

</html>