

Upgrade Postgres for a DSpace installation

Note :- These instructions previously resided in the docs folder of the DSpace distribution. It should be noted that they are very likely to be out of date at the time of reading and should not be considered the definitive instructions on how to upgrade Postgres.

1. Build new postgres.

Be sure to run configure with at least these options:

```
./configure --enable-multibyte --enable-unicode --with-java
```

2. shutdown tomcat

3. dump current data

```
pg_dumpall -o >dspace.out
```

4. shut down postgres

```
pg_ctl stop -D /dspace/database/data -m fast
```

5. back up old data directory

```
mv /dspace/database/data /dspace/database/data.old
```

6. install new postgres

7. start new postgres

```
initdb -D /dspace/database/data
```

```
edit /dspace/database/data/postgresql.conf (Add 'tcpip_socket = true')
```

```
pg_ctl start -D /dspace/database/data
```

8. restore data

```
psql -d template1 -f dspace.out
```

9. Install new JDBC driver

from the new postgres installation directory:

```
cp share/java/postgres.jar /dspace/lib
```

10. restart tomcat

Notes from postgres install docs:

If You Are Upgrading

The internal data storage format changes with new releases of PostgreSQL. Therefore, if you are upgrading an existing installation that does not have a version number "7.3.x", you must back up and restore your data as shown here. These instructions assume that your existing installation is under the "/usr/local/pgsql" directory, and that the data area is in "/usr/local/pgsql/data". Substitute your paths appropriately.

1. Make sure that your database is not updated during or after the backup.

This does not affect the integrity of the backup, but the changed data would of course not be included. If necessary, edit the permissions in the file "/usr/local/pgsql/data/pg_hba.conf" (or equivalent) to disallow access from everyone except you.

2. To back up your database installation, type:

```
pg_dumpall > outputfile
```

If you need to preserve OIDs (such as when using them as foreign keys), then use the "-o" option when running "pg_dumpall".

"pg_dumpall" does not save large objects. Check the Administrator's Guide if you need to do this.

To make the backup, you can use the "pg_dumpall" command from the version you are currently running. For best results, however, try to use the "pg_dumpall" command from PostgreSQL 7.3.1, since this version contains bug fixes and improvements over older versions. While this advice might seem idiosyncratic since you haven't installed the new version yet, it is advisable to follow it if you plan to install the new version in parallel with the old version. In that case you can complete the installation normally and transfer the data later. This will also decrease the downtime.

3. If you are installing the new version at the same location as the old one then shut down the old server, at the latest before you install the new files:

```
kill -INT `cat /usr/local/pgsql/data/postmaster.pid`
```

Versions prior to 7.0 do not have this "postmaster.pid" file. If you are using such a version you must find out the process id of the server yourself, for example by typing "ps ax | grep postmaster", and supply it to the "kill" command.

On systems that have PostgreSQL started at boot time, there is probably a start-up file that will accomplish the same thing. For example, on a Red Hat Linux system one might find that

```
/etc/rc.d/init.d/postgresql stop
```

works. Another possibility is "pg_ctl stop".

4. If you are installing in the same place as the old version then it is also a good idea to move the old installation out of the way, in case you have trouble and need to revert to it. Use a command like this:

```
mv /usr/local/pgsql /usr/local/pgsql.old
```

After you have installed PostgreSQL 7.3.1, create a new database directory and start the new server. Remember that you must execute these commands while logged in to the special database user account (which you already have if you are upgrading).

```
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data  
/usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data
```

Finally, restore your data with

```
/usr/local/pgsql/bin/psql -d template1 -f outputfile
```

using the *new* psql.

These topics are discussed at length in the Administrator's Guide, which you are encouraged to read in any case.