

MultipleDspaceOneServer

Contents

- 1 [Setting up multiple DSpace instances in one Tomcat Server.](#)
 - 1.1 [From Kyle Dykstra](#)
 - 1.2 [From Mark H. Wood](#)
 - 1.3 [From Mark Diggory](#)
 - 1.4 [From Robert Tansely](#)

Setting up multiple DSpace instances in one Tomcat Server.

From Kyle Dykstra

I had to set up a second DSpace instance that was a mirror of the original at the time the copy was made, so following is a relatively step-by-step procedure for setting up an identical DSpace instance. I called my second instance dspace-test, but as long as you maintain the same name in your config file and rename the .war files the same way, any name will do. All instances run under the same user as the original dspace, and all instances use one instance of PostgreSQL.

1. Do the following as root until told otherwise

- `sudo bash`

2. Copy [dspace-source](#)/config/dspace.cfg to dspace-test.cfg

- `cd dspace/src/dspace-1.3.2-source/config`
- `cp dspace.cfg dspace-test.cfg`

3. Edit all references of /dspace to /dspace-test

- `nano dspace-test.cfg`
- change all directory references from /dspace to /dspace-test
- change the database URL from dspace to dspace-test
- DO NOT change the database username or password

4. Make the dspace-test directory

- `mkdir /dspace-test/`

5. Own the dspace-test directory to user dspace

- `chown dspace /dspace-test/`

6. Change to user dspace

- `su dspace bash`

7. Create the dspace-test database, owned by the dspace user (Note: this step can be done with a GUI tool such as PGAdminIII, this step is also not necessary, but if a database isn't created, you will receive lots of errors when you run the fresh_install)

- `cd /usr/lib/postgresql/8.1`
- `bin/createdb --U dspace dspace-test`

8. Add the new database to pg_hba.conf so dspace can connect (as root)

- exit (changes back to user root)
- nano /etc/postgresql/8.1/main/pg_hba.conf
- Add the database dspace-test with the user dspace to the table using the IP address 127.0.0.1/32
 1. IPv4 local connections:
 - host dspace dspace 127.0.0.1/32 md5
 - host dspace-test dspace 127.0.0.1/32 md5

9. Shutdown Tomcat

- /etc/init.d/tomcat5 stop

10. Change to the [dspace-source](#) directory

- cd /dspace/src/dspace-1.3.2-source/

11. Do a fresh install using the appropriate config file

- ant clean
- ant -Dconfig=config/dspace-test.cfg fresh_install

12. Copy the [dspace-source](#)/config/dspace-test.cfg file to /dspace-test/config/dspace.cfg

- cp config/dspace-test.cfg /dspace-test/config/dspace.cfg

13. Change to the [dspace-source](#)/build directory and rename the WARs

- cd build
- mv dspace.war dspace-test.war
- mv dspace-oai.war dspace-test-oai.war

14. Copy the new WAR files to [tomcat](#)/webapps and ensure they are there

- cp ../*.war /var/lib/tomcat5/webapps/
- ls /var/lib/tomcat5/webapps/

15. Copy the dspace assetstore to dspace-test

- cp --Rv /dspace/assetstore/ /dspace-test/

16. Restart Tomcat

- /etc/init.d/tomcat5 start

17. Copy the dspace database to dspace-test

- su dspace bash
- cd /usr/lib/postgresql/8.1
- bin/pg_dump dspace > dspace
- bin/dropdb dspace-test
- bin/createdb --U dspace dspace-test
- bin/psql --d dspace-test < dspace

- - **NOTE**** This can be accomplished much easier by using a GUI tool such as PGAdminIII. Simply backup the dspace database, drop the dspace-test database, create a new database called dspace-test with user dspace, and restore the dspace-test database from this backup. The postgresql server may have to be restarted (/etc/init.d/postgresql restart) before you can drop the dspace-test database.
- 18. Rebuild the search indices

- /dspace-test/bin/index-all

19. Copy news-side.html and news-top.html so the homepage contains the same information

- cp /dspace/config/*.html /dspace-test/config/

From Mark H. Wood

Yes. With initial advice from others here I've set up a developers' environment on an old server to support multiple independent DSpace instances, so that several of us can tinker without stepping on each others' work. For a user with OS username USERNAME:

- I set up empty directories /opt/dspace-USERNAME with mask rwx,rws,rx owned by each user, with the group set to the one used by Tomcat. These will be the users' DSpace directories.
- I create an empty file 'dspace-USERNAME.war' in Tomcat's 'webapps' directory with the same ownership and mask rw,r,r. Tomcat complains about a null .war at startup, but doesn't quit. This permits the user to copy a non-null .war in without throwing the directory wide open.
- I create a PostgreSQL user USERNAME with create-database permission. pg_hba is set up to permit connections from localhost to a database by a valid user of the same name. Granting create-database lets each user 'DROP DATABASE username' and start completely over, if desired. Users may be able to create other databases, but they can't thereafter access them, which limits the fun thereof.
- One other thing that's handy is to add each developer to Tomcat's 'manager' group so they can stop/start their applications themselves instead of calling me. Unlike the other measures, this grants each developer unlimited control of **all** Tomcat applications, so be wary.

From Mark Diggory

Its all a matter of configuration, you need to setup a separate postgres database, and generate a new separate root dspace implementation and dspace war file.
For instance, if you have dspace setup as

```
DSpace Database Name: dspace
DSpace home: /dspace
DSpace war: TOMCAT_HOME/webapps/dspace.war
```

Your second configuration would need to be something like (you may want a more obvious replacement for "dspace2":

```
DSpace Database Name: dspace2
DSpace home: /dspace2
DSpace war: TOMCAT_HOME/webapps/dspace2.war
```

To attain this currently, the best thing to do is to start with your original DSpace source directory and copy/change the configuration (are you running dspace as a "dspace" user or "tomcat4" user? I'd recommend tomcat4, while others would suggest making a separate dspace user. My issue, instead of running multiple tomcats, we want to run just one, Just like the configuration for Apache HTTPD, its best to run the Tomcat engine under its own user.

1.) You'll want to create a new database:

```
createuser -U postgres -d -A -P dspace2
createdb -U dspace2 dspace2
```

2.) As Robert says, the build script can do this for you if you maintain separate dspace.cfg files for each location. For instance, heres a config with all the changes for "dspace2"

```
#
1. DSpace Configuration
#
2. revision: $revision: 1.78 $
#
3. Date: $Date: 2006/05/26 14:26:12 $
#
    a.      i.      1.      a. Basic information #####
4. DSpace installation directory
   dspace.dir = /dspace2
   *****
5. DSpace base UL. Include port number etc., but NOT trailing slash
   dspace.url = http://dspace.myu.edu:8080/dspace2
   *****
...
    a.      i.      1.      a. Database settings #####
6. Database name ("oracle", or "postgres")
   db.name = postgres
   #db.name = oracle
7. UL for connecting to database
   db.url = jdbc:postgresql://localhost:5432/dspace2
   *****
8. JDC Driver
   db.driver = org.postgresql.Driver
9. Database username and password
   db.username = dspace2
   db.password = dspace2
   *****
10. Connection pool parameters
```

3.) You'd then run:

```
ant clean
```

and then run something like

```
ant -Dconfig=/<yourinstalldir>/config/dspace2.cfg fresh_install
```

Note, these are very rough directions, approach with caution and on a test system first to make sure all is as you would like in production.

From Robert Tansely

This works no problem, I was running 4 or 5 different DSpaces on one Tomcat. Just make sure each instance has a separate install dir (esp. assetstore) and PostgreSQL database configured. Then just drop the .war files in tomcat/webapps, and you're away.

One caveat – if you're building the instances off a single source tree, be sure to do an 'ant clean' before each build/deploy, otherwise the location of the dspace.cfg to use (stuffed away in the .war WE-INF/web.xml file) might be wrong.