About DSpace

DSpace is an open source repository software package typically used for creating open access repositories for scholarly and/or published digital content. While DSpace shares some feature overlap with content management systems and document management systems, the DSpace repository software serves a specific need as a digital archives system, focused on the long-term storage, access and preservation of digital content. The first public version of DSpace was released in November 2002, as a joint effort between developers from MIT and HP Labs. Following the first user group meeting in March 2004, a group of interested institutions formed the DSpace Federation, which determined the governance of future software development by adopting the Apache Foundation's community development model as well establishing the DSpace Committer Group. In July 2007 as the DSpace user community grew larger, HP and MIT jointly formed the DSpace Foundation, a not-for-profit organization that provided leadership and support. In May 2009 collaboration on related projects and growing synergies between the DSpace Foundation and the Fedora Commons organization led to the joining of the two organizations to pursue their common mission in a not-for-profit called DuraSpace. Currently the DSpace software and user community receives leadership and guidance from DuraSpace.

Technology

DSpace is a set of cooperating Java web applications and utility programs that maintain an asset store and an associated metadata store. The web applications provide interfaces for administration, deposit, ingest, search and access. The asset store is maintained on a file system or similar storage system. The metadata, including access and configuration information is stored in a relational database and supports the use of PostgreSQL and Oracle database. DSpace currently support two primary web interfaces: JSPUI which uses JSP and the Java Servlet API and XMLUI (aka Manakin) based on Apache Cocoon, using XML and XSLT. DSpace holdings are made available primarily via a web interface, but it also supports the OAI-PMH v2.0, and is capable of exporting METS (Metadata Encoding and Transmission Standard) packages. DSpace supports the common interoperability standards used in the Institutional repository domain, such as Open Archives Initiative Protocol for Metadata Harvesting, SWORD, OpenSearch, and RSS. More recent versions of DSpace also support faceted search and browse functionality using Apache Solr.

DSpace Installation (Ubuntu 14.4 LTS)

1. **Login as root (sudo -i) than execute the following commands:**

   - Update the Ubuntu : apt-get update
   - Upgrade the Ubuntu : apt-get upgrade
   - Install OpenJDK 7 : apt-get install openjdk-7-jdk
   - Apache Maven 3.x (Java build tool) and Apache ant : apt-get install ant maven
   - Relational Database (PostgreSQL) : apt-get install postgresql
   - Servlet Engine (Apache Tomcat 7) : apt-get install tomcat7
   - apt-get install default-jdk **(Not needed if installing on Ubuntu 12.04)**

   Note: it is advisable to select from package list tomcat java server, postgresql and openssh-server during installation of Ubuntu server operating system.

2. **Create Dspace user**

useradd -m dspace
passwd dspace [enter a password for the new user dspace]
mkdir /dspace
chown dspace /dspace

3. Configure Postgresql and Create Database

- Create the PostgreSQL "dspace" user

Log in to postgresql:

sudo su postgres

Next, we will create a database called “dspace” and database user called “dspace” with password “dspace”. Don’t confuse database user with normal user. Both are different.

createuser -U postgres -d -A -P dspace

Enter password for new role: ## Enter password for the user dsapce

Enter it again: ## Re-enter password

If asked the following:

Shall the new role be allowed to create more new roles? (y/n) y

Answer "y" for yes.

Than type exit and come to root

Open up the /etc/postgresql/9.3/main/pg_hba.conf file:

nano /etc/postgresql/9.3/main/pg_hba.conf

Add the following line shown in red color at last of the file.

local all dspace md5

Type the following to restart:

/etc/init.d/postgresql restart

4. Create the PostgreSQL 'dspace' database

Login as:

sudo su dspace

createdb -U dspace -E UNICODE dspace

Than type exit and come to root

Type the following to restart postgres:

/etc/init.d/postgresql restart

NOTE: while deleting or creating the database log in to the concern user, like for dspace user (sudo su dspace) than apply the commands

5. Create Dspace directory

mkdir /build
chmod -R 777 /build
cd /build

6. Download Dsapce to /build directory

You can check latest version of Dspace from here.
Run the command mentioned below at command prompt. (Ensure that Internet is working).
wget https://github.com/DSpace/DSpace/releases/download/dspace-5.5/dspace-5.5-src-release.tar.gz

tar -zxf dspace-5.5-src-release.tar.gz
cd /build/dspace-5.5-src-release
mvn -fn package
cd dspace/target/dspace-installer
ant fresh_install

7. Configure Tomcat

nano /etc/tomcat7/server.xml

Insert the following chunk of text just above the closing </Host>

<!--Define a new context path for all DSpace web apps-->
<Context path="/xmlui" docBase="/dspace/webapps/xmlui" allowLinking="true"/>
<Context path="/sword" docBase="/dspace/webapps/sword" allowLinking="true"/>
<Context path="/oai" docBase="/dspace/webapps/oai" allowLinking="true"/>
<Context path="/jspui" docBase="/dspace/webapps/jspui" allowLinking="true"/>
<Context path="/solr" docBase="/dspace/webapps/solr" allowLinking="true"/>

Than close the file

OR

You may use the below method also to configure the Tomcat instead of inserting the above text in server.xml file

copy any web applications from /dspace/webapps/ to the appropriate place for your servlet container. For example, '$CATALINA_HOME/webapps' for Tomcat.

First set the environment variables to Tomcat server.

Edit file /etc/profile,

nano /etc/profile

Add the following lines at the end:

export CATALINA_BASE=/var/lib/tomcat7
export CATALINA_HOME=/usr/share/tomcat7

Save and close the file. Then, run the following command to take effect the environment variables settings.

source /etc/profile

Now, copy the dspace/webapps directory contents to the tomcat webapps directory.

sudo cp -r /dspace/webapps/* $CATALINA_BASE/webapps/

8. Java environment settings for Tomcat webapp server/ JVM memory (heap) setting

nano /etc/default/tomcat7

```
# You may pass JVM startup parameters to Java here. If unset, the default
# options will be: -Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC
JAVA_OPTS="-Djava.awt.headless=true –Xmx768m –Xms128m -XX:MaxPermSize=1024m"

# Use "-XX:+UseConcMarkSweepGC" to enable the CMS garbage collector (improved
# response time). If you use that option and you run Tomcat on a machine with
# exactly one CPU chip that contains one or two cores, you should also add
# the "-XX:+CMSIncrementalMode" option.
JAVA_OPTS="-Djava.awt.headless=true -Xmx512m -XX:+UseConcMarkSweepGC"
```

nano /etc/init.d/tomcat7

```
# Default Java options
# Set java.awt.headless=true if JAVA_OPTS is not set so the
# Xalan XSL transformer can work without X11 display on JDK 1.4+
# It also looks like the default heap size of 64M is not enough for most cases
# so the maximum heap size is set to 128M
if [ -z "$JAVA_OPTS" ]; then
        JAVA_OPTS="-Djava.awt.headless=true -Xmx1024M"
```

9. Java environment settings for other java web applications

nano /etc/environment

Add the below line in the file:

```
JAVA_HOME="/usr/lib/jvm/default-java"
JAVA_OPTS="-Djava.awt.headless=true -Xmx1024m -Xms512m -Dfile.encoding=UTF-8"
```

Fix Tomcat permissions, and restart the Tomcat server

```
chown tomcat7:tomcat7 /dspace -R
```

Restart Tomcat

```
/etc/init.d/tomcat7 restart
```

10. Make an initial administrator account (an e-person) in DSpace:

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

It will ask to enter email address for user login. Enter an email address (e.g. dspace@localhost).
Enter First name and surename (e.g. dspace)
Enter a password.

11. View the installed default DSpace interface.

You can load either one Dspace interface in a browser.
http://localhost:8080/xmlui
http://localhost:8080/jspui

Notes / Miscellaneous help and Tips & Tricks on DSpace

1. If you want to uninstall postgresql and any related package, you should use autoremove:

sudo apt-get --purge autoremove postgresql*

2. Commands to get the statistics:

In regards to statistics (e.g. /xmlui/statistics/ and /jspui/statistics/), I've figured this one out! One needs to use the commands "[dspace]/bin/dspace stat-initial" and "[dspace]/bin/dspace stat-report-initial" to generate the traditional pre-1.6 statistics. Then use the "stat-monthly","stat-report-monthly","stat-general", and **"stat-report-general"** commands as cronjobs moving forward for regular updates. [dspace]/bin/dspace item-counter (in dspace.cfg configure file : webui.strengths.show = **true** , webui.strengths.cache = **true**)

3. DSpace Handle server Configuration

The Handle Server

First a few facts to clear up some common misconceptions:

- You don't **have** to use CNRI's Handle system. At the moment, you need to change the code a little to use something else (e.g PURLs) but that should change soon.
- You'll notice that while you've been playing around with a test server, DSpace has apparently been creating handles for you looking like *hdl:123456789/24* and so forth. These aren't really Handles, since the global Handle system doesn't actually know about them, and lots of other DSpace test installs will have created the same IDs. They're only really Handles once you've registered a prefix with CNRI (see below) and have correctly set up the Handle server included in the DSpace distribution. This Handle server communicates with the rest of the global Handle infrastructure so that anyone that understands Handles can find the Handles your DSpace has created.
  If you want to use the Handle system, you'll need to set up a Handle server. This is included with DSpace. Note that this is not required in order to evaluate DSpace; you only need one if you are running a production service. You'll need to obtain a Handle prefix from the central CNRI Handle site.

A Handle server runs as a separate process that receives TCP requests from other Handle servers, and issues resolution requests to a global server or servers if a Handle entered locally does not correspond to some local content. The Handle protocol is based on TCP, so it will need to be installed on a server that can broadcast and receive TCP on port 2641. If your DSpace server sits behind a firewall, also ensure that port 2641 is opened on your firewall as well, both for udp and tcp traffic.

1. To configure your DSpace installation to run the handle server, run the following command:
   [dspace]/bin/dspace make-handle-config [dspace]/handle-server
2. Ensure that *[dspace]/handle-server* matches whatever you have in *dspace.cfg* for the *handle.dir* property.

3. Edit the resulting *[dspace]/handle-server/config.dct* file to include the following lines in the *"server_config"*clause:

   "storage_type" = "CUSTOM"
   "storage_class" = "org.dspace.handle.HandlePlugin"

4. This tells the Handle server to get information about individual Handles from the DSpace code.

5. Once the configuration file has been generated, you will need to go to http://hdl.handle.net/4263537/5014 to upload the generated sitebndl.zip file. The upload page will ask you for your contact information. An administrator will then create the naming authority/prefix on the root service (known as the Global Handle Registry), and notify you when this has been completed. You will not be able to continue the handle server installation until you receive further information concerning your naming authority.

6. When CNRI has sent you your naming authority prefix, you will need to edit the *config.dct* file. The file will be found in */[dspace]/handle-server*. Look for *"300:0.NA/YOUR_NAMING_AUTHORITY"*. Replace *YOUR_NAMING_AUTHORITY* with the assigned naming authority prefix sent to you.

7. Now start your handle server (as the dspace user):

   [dspace]/bin/start-handle-server

8. Note that since the DSpace code manages individual Handles, administrative operations such as Handle creation and modification aren't supported by DSpace's Handle server.

Updating Existing Handle Prefixes

If you need to update the handle prefix on items created before the CNRI registration process you can run the *[dspace]/bin/dspace update-handle-prefix script*. You may need to do this if you loaded items prior to CNRI registration (e.g. setting up a demonstration system prior to migrating it to production). The script takes the current and new prefix as parameters. For example:

[dspace]/bin/dspace update-handle-prefix 123456789 1303

This script will change any handles currently assigned prefix 123456789 to prefix 1303, so for example handle 123456789/23 will be updated to 1303/23 in the database.

4. Discovery Solr Index Maintenance

Command used:	[dspace]/bin/dspace index-discovery [-cbhf[r <item handle>]]
Java class:	org.dspace.discovery.IndexClient
Arguments (short and long forms):	Description
	called without any options, will update/clean an existing index
-b	(re)build index, wiping out current one if it exists
-c	clean existing index removing any documents that no longer exist in the db
-f	if updating existing index, force each handle to be reindexed even if uptodate

-h	print this help message
-o	optimize search core
-r <item handle>	remove an Item, Collection or Community from index based on its handle

[dspace]/bin/dspace index-discovery -o

(It is strongly recommended to run maintenance on the Discovery Solr index daily (from crontab or your system's scheduler), to prevent your servlet container from running out of memory)

5. Thumbnail Creation

To create thumbnail of images etc run the command:

/dspace/bin/dspace filter-media

Available Command-Line Options:

- **Help** : [dspace]/bin/dspace filter-media -h
  - Display help message describing all command-line options.
- **Force mode** : [dspace]/bin/dspace filter-media -f
  - Apply filters to ALL bitstreams, even if they've already been filtered. If they've already been filtered, the previously filtered content is overwritten.
- **Identifier mode** : [dspace]/bin/dspace filter-media -i 123456789/2
  - Restrict processing to the community, collection, or item named by the identifier - by default, all bitstreams of all items in the repository are processed. The identifier must be a Handle, not a DB key. This option may be combined with any other option.
- **Maximum mode** : [dspace]/bin/dspace filter-media -m 1000
  - Suspend operation after the specified maximum number of items have been processed - by default, no limit exists. This option may be combined with any other option.
- **No-Index mode** : [dspace]/bin/dspace filter-media -n
  - Suppress index creation - by default, a new search index is created for full-text searching. This option suppresses index creation if you intend to run index-update elsewhere.
- **Plugin mode** : [dspace]/bin/dspace filter-media -p "PDF Text Extractor","Word Text Extractor"
  - Apply ONLY the filter plugin(s) listed (separated by commas). By default all named filters listed in the *filter.plugins* field of *dspace.cfg* are applied. This option may be combined with any other option. *WARNING:* multiple plugin names must be separated by a comma (i.e. ',') and NOT a comma followed by a space (i.e. ', ').
- **Skip mode** : [dspace]/bin/dspace filter-media -s 123456789/9,123456789/100
  - SKIP the listed identifiers (separated by commas) during processing. The identifiers must be Handles (not DB Keys). They may refer to items, collections or communities which should be skipped. This option may be combined with any other option. *WARNING:* multiple identifiers must be separated by a comma (i.e. ',') and NOT a comma followed by a space (i.e. ', ').
  - NOTE: If you have a large number of identifiers to skip, you may maintain this comma-separated list within a separate file (e.g. *filter-skiplist.txt*). Use the following format to call the program. *Please note the use of the "grave" or "tick" (`) symbol and do not use the single quotation.*
    - [dspace]/bin/dspace filter-media -s `less filter-skiplist.txt`
- **Verbose mode** : [dspace]/bin/dspace filter-media -v
  - Verbose mode - print all extracted text and other filter details to STDOUT.

Adding your own filters is done by creating a class which *implements* the org.dspace.app.mediafilter.FormatFilter interface. See the Creating a new Media/Format Filter topic and comments in the source file FormatFilter.java for more information. In theory filters could be implemented in any programming language (C, Perl, etc.) However, they need to be invoked by the Java code in the Media Filter class that you create.

6. Nightly Cronjob Setup

- To schedule these scripts to run, execute the command crontab –e as the DSpace user
  - Then add the following lines:
  - # Send out subscription e-mails at 01:00 every day
  - 0 1 * * * [dspace]/bin/sub-daily
  - # Run the media filter at 02:00 every day
  - 0 2 * * * [dspace]/bin/filter-media
  - # Run the checksum checker at 03:00
  - 0 3 * * * [dspace]/bin/checker -lp
  - # Mail the results to the sysadmin at 04:00
  - 0 4 * * * [dspace]/bin/dsrun org.dspace.checker.DailyReportEmailer –c
  - # to prevent your servlet container from running out of memory at 05:00
  - 0 5 * * * [dspace]/bin/dspace index-discovery -o
  - # Item counting of collections (in dspace.cfg configure : webui.strengths.show = true , webui.strengths.cache = true)
  - 0 6 * * * [dspace]/bin/dspace item-counter

PostgreSQL also benefits from regular ’vacuuming‘

- # Clean up the database nightly at 4.20am
- 20 4 * * * vacuumdb --analyze dspace > /dev/null 2>&1

Generate Statistical reports:

- # Run stat analyses
- 30 1 * * * [dspace]/bin/stat-general
- 45 1 * * * [dspace]/bin/stat-monthly
- 30 2 * * * [dspace]/bin/stat-report-general
- 45 2 * * * [dspace]/bin/stat-report-monthly

7. Some useful settings in dspace.cfg file

- # to remove restriction for not to upload file
- webui.submit.upload.required = true (Uncomment and Change to false)
- # Item counting of collections

- webui.strengths.show = true , webui.strengths.cache = true
- Transforming DSpace Content (MediaFilters) https://wiki.duraspace.org/pages/viewpage.action?pageId=27001634

8. How to configure embargo in DSpace

- The first step is to create the following variable in the dspace.cfg file under the Embargo Setting section

Variable for simple (false) and advance (true) embargo settings
```
webui.submission.restrictstep.enableAdvancedForm=false
```

- **Next step is to modify *item-submission.xml* given under the /dspace/config/ folder**

To enable the new embargo, changes are required to the *item-submission.xml* file, located in your config directory. This file determines which steps are executed in the submission of a new item.

Two new submission steps have been introduced in the file. By default, they are not activated yet:

- *AccessStep*: the step in which the user can set the embargo at item level, effectively restricting access to the item metadata.
- *UploadWithEmbargoStep*: the step in which the user can set the embargo at bitstream level. **If this step is enabled, the old *UploadStep* must be disabled. Leaving both steps enabled will result in a system failure.**

Here is an extract from the new file: **(I activated Step 4 only)**

```
<!--Step 3 will be to Manage Item access.
       <step>
           <heading>submit.progressbar.access</heading>
           <processing-class>org.dspace.submit.step.AccessStep</processing-class>
           <jspui-binding>org.dspace.app.webui.submit.step.JSPAccessStep</jspui-
binding>
           <xmlui-
binding>org.dspace.app.xmlui.aspect.submission.submit.AccessStep</xmlui-binding>
           <workflow-editable>true</workflow-editable>
       </step>
       -->

<!-- Step 4 Upload Item with Embargo Features (not supported in JSPUI)
            to enable this step, please make sure to comment-out the previous step
"UploadStep"
       <step>
           <heading>submit.progressbar.upload</heading>
           <processing-class>org.dspace.submit.step.UploadWithEmbargoStep</processing-
class>
           <jspui-
binding>org.dspace.app.webui.submit.step.JSPUploadWithEmbargoStep</jspui-binding>
           <xmlui-
binding>org.dspace.app.xmlui.aspect.submission.submit.UploadWithEmbargoStep</xmlui-
binding>
           <workflow-editable>true</workflow-editable>
       </step>
        -->
```

To enable the new Embargo, ensure that the new steps are uncommented and the old UploadStep is commented out.

Simple Embargo Settings

Using the simple embargo settings, submitters will be able to define embargoes bound to specific dates, that are applied to all anonymous and default read access. To keep the interface simple, options to apply embargoes for particular groups of DSpace users are not shown. The simple embargo settings interface assumes that embargoes always start immediately upon submission, so only end dates are configurable.

AccessStep

The simple *AccessStep* Embargo form renders three options for the user:

- *Private item*: to hide an item's metadata from all searches and browse indexes, as well as external interfaces such as OAI-PMH.
- *Embargo Access until Specific Date*: to indicate a date until which the item will be embargoed.
- *Reason*: to elaborate on the specific reason why an item is under embargo.

When Embargo is set, it applies to Anonymous or to any other Group that is indicated to have *default read access* for that specific collection.

9. Dspace Upgradation from old version to new version Tips & Tricks

Install a fresh installation of the latest version of Dspace as per installation instructions, and then simply upgrade your data (primarily the database). This is a much easier multi-version upgrade process; however it does not automatically upgrade all of your configurations or customizations (as you'll be migrating to a fresh, uncustomized installation). So, depending on how complex (or important) your customizations were, this may not fully meet your needs.

Install the version of DSpace you wish to upgrade to by following the latest installation instructions For example to upgrade to DSpace 4.x follow the instructions at Installing DSpace. At this point, you'll have a fresh copy of DSpace, but no data. Make a backup copy of your old DSpace database. You'll use this to upgrade your data to the latest version of DSpace:

For example, in PostgreSQL use the "pg_dump" command to backup all your old data:

pg_dump -U [database-user] -f my-dspace-db-backup.sql [database-name]

Create an exact replica of your old DSpace database (this is mostly just for safety purposes – you don't want to lose any of your Production data!)

For example, in PostgreSQL use the "createdb" and "psql" commands:

Create a replica database. I've called this one "dspace-upgrade" just as an example

createdb -U [database-user] -E UNICODE dspace-upgrade

Load the old data into this replica database

psql -U [database-user] -f my-dspace-db-backup.sql dspace-upgrade

Now, upgrade this replica database to be compatible with the version of DSpace you wish to upgrade to. This is done utilizing the "database_schema-*.sql" upgrade scripts provided in the [dspace-source]/dspace/etc/postgres/ folder.

For example, suppose you are upgrading from DSpace 1.2.x to 4.x. That means you'll be upgrading your database from being 1.5.x compatible to being 4.x compatible. In order to do that, you will run all these database upgrade scripts in order (again this example is just for PostgreSQL):

```
# Upgrade 'dspace-upgrade' database from 1.2.x to 1.3.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_12-13.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.3.x to 1.4.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_13-14.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.4.x to 1.5.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_14-15.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.5.x to 1.6.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_15-16.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.6.x to 1.7.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_16-17.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.7.x to 1.8.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_17-18.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.8.x to 3.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_18-3.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 3.x to 4.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_3-4.sql dspace-upgrade
```

You now have a 4.x compatible database (in "dspace-upgrade")! So, you can dump its data out to a file:

For example, to dump "dspace-upgrade" to a file named "my-dspace-upgraded-db.sql":

pg_dump -U [database-user] -f my-dspace-upgraded-db.sql dspace-upgrade **(log into as dspace user and save the dump in /home/dspace folder)**

Remember your "fresh" installation of the newer version of DSpace (in this case DSpace 4.x)? Well, now you will just move your old data over to that fresh installation. First, you need to move over your upgraded Database data. That's done by removing the default "fresh installation" database, and replacing it with your upgraded version:

```
# Delete the default (MAKE SURE IT IS EMPTY) "fresh install" Database on the NEW DSpace

dropdb -U [database-user] [database-name]   (log into as dspace user (sudo su dspace) to delete the database)
```

Recreate an empty DB
createdb -U [database-user] -E UNICODE [database-name] **(log into as dspace user (sudo su dspace) to create database)**

Load in the newly upgrade data (from "my-dspace-upgraded-db.sql")

psql -U [database-user] -f my-dspace-upgraded-db.sql [database-name] **(log into as dspace user (sudo su dspace) to restore the database and take backup)**

Next, copy over your old DSpace "Assetstore" directory into the newly installed version of DSpace. This assetstore directory contains all the actual content files

cp -R [path-to-old-dspace]/assetstore/* [path-to-new-dspace]/assetstore/

Finally, reindex all of your DSpace content in the newly installed version of DSpace. For example, assuming you are using Discovery faceted/filtered search/browse (the default in DSpace 4.x), you'd run:

[dspace]/bin/dspace index-discovery –f

At this point, you should have a fresh installation of a newer version of DSpace with your content migrated to it. As a final step, you may wish to re-configure or re-customize the fresh installation (based on your old settings). Also be sure to review all the Upgrade Instructions for the versions you "skipped over". Sometimes there are important notes or details in there!

10. Command in case facing problem in deletion of records after data migration

- **If not able to delete item records from the collection please run the below command**

[dspace]/bin/dspace index-lucene-init -r -t –v

- **If not able to delete whole collection use the following commands**

While updating you get some errors like that:

psql:database_schema_15-16.sql:105: ERROR: constraint
"community2collection_collection_id_fkey" of relation
"community2collection" does not exist
ALTER TABLE
psql:database_schema_15-16.sql:108: ERROR: constraint
"community2community_child_comm_id_fkey" of relation
"community2community" does not exist
ALTER TABLE
psql:database_schema_15-16.sql:111: ERROR: constraint
"collection2item_item_id_fkey" of relation "collection2item" does not exist

If so, inspect your database and use the references for the foreign keys
to delete them, e.g::

psql [YourDBName]
\d [TableName]
will show you the foreign keys, e.g.:
\d community2collection

```
Table "public.community2collection"
    Column     |  Type   | Modifiers
---------------+---------+-----------
id             | integer | not null
community_id   | integer |
collection_id | integer |
Indexes:
    "community2collection_pkey" PRIMARY KEY, btree (id)
    "community2collection_collection_id_idx" btree (collection_id)
    "community2collection_community_id_idx" btree (community_id)
Foreign-key constraints:
    "$1" FOREIGN KEY (community_id) REFERENCES community(community_id)
    "$2" FOREIGN KEY (collection_id) REFERENCES collection(collection_id)
```

In that case you got to run:

ALTER TABLE collection2item DROP CONSTRAINT "$2";
ALTER TABLE community2community DROP CONSTRAINT "$2";
ALTER TABLE community2collection DROP CONSTRAINT "$2";

The reason for this is:

While updating the database it can happen that not all the commands in
the database_schema_xx-xx.sql are executed properly.

The commands to drop foreign keys:

ALTER TABLE collection2item DROP CONSTRAINT collection2item_item_id_fkey;

ALTER TABLE community2community DROP CONSTRAINT
community2community_child_comm_id_fkey;

ALTER TABLE community2collection DROP CONSTRAINT
community2collection_collection_id_fkey;

assume that the keys got default names, whereas depending on the database version and operating system on which your instance started,they can have no names and just be counted internally.

If you got no names or other names the script will skip the commands, with the above mentioned ERROR.

This will lead to errors running DSpace as now 2 constraints exist and one is unable to delete items, collections etc.

11. Changing Port (Please note you need to change ports at various other files also)

Allow Tomcat to listen on ports "80" and "443"
Setup "authbind" for Tomcat

To enable Tomcat to listen on a privileged port below 100, we need to enable "authbind". Edit the /etc/default/tomcat6 file as follows:

sudo nano /etc/default/tomcat7

Remove the hash sign from in front of the authbind parameter and change authbind to yes as follows

```
# If you run Tomcat on port numbers that are all higher than 1023, then you
# do not need authbind.  It is used for binding Tomcat to lower port numbers.
# NOTE: authbind works only with IPv4.  Do not enable it when using IPv6.
# (yes/no, default: no)
AUTHBIND=yes
```

NANO Editor Help

```
CTL+O       = Save the file and then press Enter
CTL+X       = Exit "nano"
CTL+K       = Delete line
CTL+U       = Undelete line
CTL+W       = Search for $string$
CTL+\       = Search for and replace $string$
ALT+C       = Show line numbers
```

More info = http://en.wikipedia.org/wiki/Nano_(text_editor)

Now we need to tell "authbind" that Tomcat is allowed to use lower port numbers. Type the following commands:

```
sudo touch /etc/authbind/byport/80

sudo touch /etc/authbind/byport/443

sudo chmod 0755 /etc/authbind/byport/80

sudo chmod 0755 /etc/authbind/byport/443

sudo chown tomcat7.tomcat7 /etc/authbind/byport/80

sudo chown tomcat7.tomcat7 /etc/authbind/byport/443

cd /etc/authbind/byport

ls -l
```

Now Tomcat has permission to use ports 80 and 443. See below for an example listing of the files in the /etc/authbind/byport folder.

```
root@ir1:/etc/authbind/byport# ls -l
total 0
-rwxr-xr-x 1 tomcat6 tomcat6 0 2011-06-10 18:33 443
-rwxr-xr-x 1 tomcat6 tomcat6 0 2011-06-10 18:33 80
```

Setup Tomcat for open port 80

Now we tell the Tomcat server to listen on the "authbind" ports. Edit the following file.

```
sudo nano /etc/tomcat7/server.xml
```

Find the connector for port 8080 and change it to port 80.

See example below.

```
<Connector port="80" protocol="HTTP/1.1"
           enableLookups="false"
          connectionTimeout="20000"
          URIEncoding="UTF-8"
          redirectPort="443" />
```

References:

1. http://en.wikipedia.org/wiki/DSpace
2. http://www.dspace.org/