
# Dspace Implementation Documentation
# Version 2.0

Last Modified: May 1, 2007
Modified by Maike Dulk

## Versions and Modifications

| Revision |
|---|
| 2.0 |

| Changes made to this document | | | |
|---|---|---|---|
| **Date** | **Modified by** | **Version** | **Changes** |
| July 21, 2005 | Shaun Lum (shaunlum@uvic.ca) | 1.0 | Document Draft |
| April 30, 2007 | Maike Dulk (maike@uvic.ca) | 2.0 | Lots |

# Table of Contents

# About DSpace

A groundbreaking digital repository system, DSpace captures, stores, indexes, preserves, and redistributes an organization's research material in digital formats.

## Descriptive Metadata

Each Item has one qualified Dublin Core metadata record. The set of elements and qualifiers used by MIT Libraries is the default configuration included in the DSpace source code. These are loosely based on the Library Application Profile set of elements and qualifiers, though there are some differences.

## Administrative Metadata

This includes preservation metadata, provenance and authorization policy data. Most of this is held within DSpace's relation DBMS schema. Provenance metadata (prose) is stored in Dublin Core records. Additionally, some other administrative metadata (for example, bitstream byte sizes and MIME types) is replicated in Dublin Core records so that it is easily accessible outside of DSpace.

## Structural Metadata

This includes information about how to present an item, or bitstreams within an item, to an end-user, and the relationships between constituent parts of the item. As an example, consider a thesis consisting of a number of TIFF images, each depicting a single page of the thesis. Structural metadata would include the fact that each image is a single page, and the ordering of the TIFF images/pages. Structural metadata in DSpace is currently fairly basic; within an item, bitstreams can be arranged into separate bundles as described above. A bundle may also optionally have a primary bitstream. This is currently used by the HTML support to indicate which bitstream in the bundle is the first HTML file to send to a browser.

## Bitstreams Storage

All uploaded files of dspace are located on neptune at *`$DSPACE/assetstore/`*

**How files are stored**
Stores are numbered, starting with zero, then counting upwards. Each bitstream entry in the database has a store number, used to retrieve the bitstream when required.
At the moment, the store in which new bitstreams are placed is decided using a configuration parameter, and there is no provision for moving bitstreams between stores. Administrative tools for manipulating bitstreams and stores will be provided in future releases. Right now you can move a whole store (e.g. you could move store number 1 from `/localdisk/store` to `/fs/anotherdisk/store` but it would still have to be store number 1 and have the exact same contents.

Bitstreams also have an 38-digit internal ID, different from the primary key ID of the bitstream table row. This is not visible or used outside of the bitstream storage manager. It is used to determine the exact location (relative to the relevant store directory) that the bitstream is stored in traditional or SRB storage. The first three pairs of digits are the directory path that the bitstream is stored under. The bitstream is stored in a file with the internal ID as the filename.

For example, a bitstream with the internal ID `12345678901234567890123456789012345678` is stored in the directory:
`(assetstore dir)/12/34/56/12345678901234567890123456789012345678`

The reasons for storing files this way are:

- Using a randomly-generated 38-digit number means that the 'number space' is less cluttered than simply using the primary keys, which are allocated sequentially and are thus close together. This means that the bitstreams in the store are distributed around the directory structure, improving access efficiency.

- The internal ID is used as the filename partly to avoid requiring an extra lookup of the filename of the bitstream, and partly because bitstreams may be received from a variety of operating systems. The original name of a bitstream may be an illegal UNIX filename.

### Backing up DSpace files.

To back up all of the bitsream files, copy the `$DSPACE/assetstore` folder to a secure location.

## Upgrading DSpace

The general upgrading process can be found in
`$DSPACE_SRC/docs/update.html`
or
`http://dspace.org/technology/system-docs/update.html`
(althought this web page is NOT up to date – it misses the instructions for updateing to 1.4.1 at the time of writing, April 2007)

### Uvic specifics

#### update log from 1.2.2 to 1.3.2

- backed up $DSPACE
- backed up $CATALINA_HOME/webapps
- backed up dspace database:
  `/usr/local/pgsql/bin/pg_dump dspace > dspace_dump`
  and copy/scp this file over to a safe place
- cleaned out `$DSPACE/jsp/local` by removing all garbage: files that are actually not localised at all, old backup files etcetera
- copied this directory over to 1.3.2 `DSPACE_SRC` directory
- diffed all the remaining jsp's with the 1.3.2 ones and updated the localised ones
- display-item.jsp:
  copied the creativecommons paragraph over
- advanced.jsp:
  copied the extra search terms over
  adapted the `fmt:message key=` mechanism
- header_default.jsp:
  made something better of the logo / text anarchy and created a new logo
- Copy the PostgreSQL driver JAR to the source tree
  `cd *$DSPACE/*lib`
  `cp postgresql.jar $DSPACE_SRC/lib`
- Remove the old version of xerces.jar from your installation, so it is not inadvertently later used:
  `rm [dspace]/lib/xerces.jar`
- Installed the new config files by moving `dstat.cfg` and `dstat.map from $DSPACE_SRC/config/ to $DSPACE/config/`
- Added new parameters to $DSPACE/dspace.cfg:

```
###### Statistical Report Configuration Settings ######

# should the stats be publicly available?  should be set to false if you only
# want administrators to access the stats, or you do not intend to generate
# any
report.public = false

# directory where live reports are stored
report.dir = /dspace/reports/
```

- repaired the full text search by removing the error in that paragraph of dspace.cfg
- stop Tomcat: `$CATALINA_HOME/bin/shutdown.sh`
- build DSPACE with:
  `cd $DSPACE_SRC`
  `ant -Dconfig=$DSPACE/config/dspace.cfg update`
- Updated the database
  `/usr/local/pgsql/bin/psql -f &DSPACE_SRC/etc/database_schema_12-13.sql dspace -h localhost`

- copied the .war in `$DSPACE_SRC/build` to Tomcat webapps sub-directory:
  `cp $DSPACE_SRC/build*.war $CATALINA_HOME/webapps`
- deleted the `$CATALINA_HOME/webapps/dspace` and `/dspace-oai` directories
- restarted Tomcat : `$CATALINA_HOME/bin/startup.sh`

**update log from 1.3.2 to 1.4.1**

- backed up $DSPACE
- backed up $CATALINA_HOME/webapps
- backed up dspace database:
  `/usr/local/pgsql/bin/pg_dump dspace > dspace_dump2`
  and copy/scp this file over to a safe place
- copied `$DSPACE/jsp/local` over to 1.4.1 `$DSPACE_SRC` directory
- diffed the localised jsp's with the 1.4.1 ones and updated the localised ones
- Copy the PostgreSQL driver JAR to the source tree
  `cd *$DSPACE*/lib`
  `cp postgresql.jar $DSPACE_SRC/lib`
- Downloaded the java 1.4 versions of the **bouncycastle** (no, I did **not** make this up) libraries from **http://www.bouncycastle.org/latest_releases.html** and copied them to `$DSPACE/lib`
- the input-forms.xml **field tags** needed an additional element:
  `<dc-schema>dc</dc-schema>`
  Added this to all elements.
- Created a sciences index file `uvsi.xml` for the Uvic based on the former '*controlled vocabularies'* HTML page, put that in `$DSPACE/config/controlled-vocabularies/`
- to enable this, add `<vocabulary>uvsi</vocabulary>` to the appropriate tag
- copy the `$DSPACE/config/controlled-vocabularies/` directory and `$DSPACE/config/input-forms.xml` over to the $DSPACE directory
- There are many changes in the `dspace.cfg` file between 1.3.2 and 1.4.1
  Instead of adding the new paragraphs to it as the manual states, I thought it better to diff the old and new versions and merge them.
- enabled the controlled vocabulary mechanism in `dspace.cfg` (removed the #)
  `webui.controlledvocabulary.enable = true`
- build DSPACE with:
  `cd $DSPACE_SRC`
  `ant -Dconfig=$DSPACE/config/dspace.cfg update`
- `Updated the database`
  `/usr/local/pgsql/bin/psql -f &DSPACE_SRC/etc/database_schema_13-14.sql dspace -h localhost`
- `initialised the stat reports by running these scripts:`
  `$DSPACE/bin/stat-general`
  `$DSPACE/bin/stat-initial`
  `$DSPACE/bin/stat-monthly`
  `$DSPACE/bin/stat-report-general`
  `$DSPACE/bin/stat-report-initial`
  `$DSPACE/bin/stat-report-monthly`
- Rebuilded the search index:
  `$DSPACE/bin/index-all`
- stop Tomcat: `$CATALINA_HOME/bin/shutdown.sh`
- copied the .war in `$DSPACE_SRC/build` to Tomcat webapps sub-directory:
  `cp $DSPACE_SRC/build*.war $CATALINA_HOME/webapps`
- deleted the `$CATALINA_HOME/webapps/dspace` and `/dspace-oai` directories
- restarted Tomcat : `$CATALINA_HOME/bin/startup.sh`

<need to research this>
$DSPACE_SRC/src/org/dspace/app/oai/ETDMSCrossWalk.java
$DSPACE_SRC/src/org/dspace/app/oai/ETDMSCrossWalk.class

## Uvic Customisations

- display-item.jsp:
  - added Creative Commons paragraph at the bottome

- layout/header-default.jsp:
  - changed the table in the header to include localised images and hyperlinks

- search/advanced.jsp:
  - Added two search fields – in all three locations in the file:

```
<option value="department" <%= field1.equals("department") ?
"selected=\"selected\"" : "" %>><fmt:message
key="jsp.search.advanced.type.department"/></option>
<option value="copyright" <%= field1.equals("copyright") ?
"selected=\"selected\"" : "" %>><fmt:message
key="jsp.search.advanced.type.copyrightdate"/></option>
```

- added these new search indexes in `config/dspace.cfg.`

- heavily costomised the `config/input-forms.xml`

- created a custom UVic sciences index file `config/controlled-vocabularies/uvsi.xml` and enabled this in `config/input-forms.xml`

ItemTag.java
- Display of records changed meta data
- Added supervisor, department, degree, copyright date

Browse by date shows records with copyright dates instead of date of issue
- ItemListTag.java
Creative commons image and link at bottom of all records.

# Dspace Open Archives Initiatives(OAI)

The dspace OAI allows for metadata harvesters such as LAC or CARL to harvest UvicDSpace. The URL that harvesters use is http://dspace.library.uvic.ca:8080/dspace-oai/request

## DSpace resources

http://dspace.org - Dspace Home page
http://www.openarchives.org/ -

# Household stuff

## Terms and Definitions

**$DSPACE**
Refers to the install directory of the existing Dspace installation.
*/usr/local/dspace/*

**$DSPACE_SRC/**
Refers to the source directory for DSpace 1.2.2.
*/usr/local/src/dspace/dspace-1.2.2-source/*

**$CATALINA_HOME/**
Refers to the home directory of Tomcat
*/usr/local/tomcat*

**OAI**
The Open Archives Initiative has developed a protocol for metadata harvesting. This allows sites to programmatically retrieve or 'harvest' the metadata from several sources, and offer services using that metadata, such as indexing or linking services. Such a service could allow users to access information from a large number of sites from one place.

**WAR**
**W**eb **Ar**chive. Zipped JAR-type archive file that contains all build files including class, xml, and jsp files of DSpace

## Locations

DSpace base URL
*https://dspace.library.uvic.ca:8443/dspace/*

DSpace public URL (redirection to https://dspace.library.uvic.ca:8443/dspace)
*http://dspace.library.uvic.ca*

DSpace login URL https://dspace.library.uvic.ca:8443/dspace/password-login

DSpace admin *https://dspace.library.uvic.ca:8443/dspace-admin*

DSpace OAI http://dspace.library.uvic.ca:8080/dspace-oai /request

DSpace source and install directories, plus the postgesql database are stored on neptune (Mac Server).

## Accessing DSpace on Neptune

You can access DSpace on Neptune through any basic terminal by typing
`ssh dspace@neptune`
– type in neptune dspace UNIX password

## Dspace basic layout

**Install Directory $DSPACE**
`/usr/local/dspace`

**Source Directory $DSPACE_SRC**
`/usr/local/src/dspace/dspace-<version number>-source`

**DSpace.cfg**
`$DSPACE/config/dspace.cfg`

**input-forms.xml**

`$DSPACE/config/input-forms.xml`

**DSpace log files**
`$DSPACE/log/dspace.log`

`default.license` – default license that users must grant when submitting items
`mediafilter.cfg` – Media Filter configuration
`news-side.html` – Text of the front-page news in the sidebar
`news-top.html` – Text of the front-page news in the top box
`emails/` - Texts of emails sent out by the system

## Tomcat

**Webapps directory : l**ocation of all files that need to run a webpage
`$CATALINA_HOME /webapps`

**Bin directory : l**ocation of the tomcat start and stop executables
`$CATALINA_HOME/bin`

**Tomcat log files**
`$CATALINA_HOME/log`

## Debugging and Logs

**Logs**
A useful tool when working with DSpace is the log file which is located at `$DSPACE/log`.
You can view the entire log normally using `less`, or you can use the command listed below.

`tail –n` **<number>** `$DSPACE/log/dspace.log`

It will allow you to see the last **number** lines of the log at runtime of DSpace.
Tomcat logs arelocated in `$CATALINA_HOME/logs`

## Starting and Stopping the Tomcat Server

To start Tomcat:
`$CATALINA_HOME/bin/startup.sh`

To stop Tomcat:
`$CATALINA_HOME/bin/shutdown.sh`

To check to see if tomcat is running, type ps and there shouldn't be a tomcat process running in the background.

## Accessing the Postgresql Database

You can use any suitable db admin tool phpmyAdmin to access the postgresql database.
The easiest way of accessing it is to create an ssh tunnel to neptune so that you do not have to fiddle with the pg_hba.conf settings.

To do this, open a console and enter:
`ssh -L 5433:localhost:5432 neptune -l dspace`
and give the neptune dspace password

Create a connection to localhost and port 5433 (instead of the default 54323),
loginname: **dspace**
password: **dspace DATABASE password**
host: **localhost**
port: **5433**
database: **dspace**