DSpace Technical Notes
Nason Bimbe
November 2015

Purpose
To provide the people that will be tasked to managing and maintenance of the technical platform hosting the RNRA Digital Repository. The DR is a part of the wider Knowledge and Information Management System (KIMS).

Target Audience
- ICT personnel
- Anyone with ICT knowledge and interested in Digital Repositories

Learning Outcomes
The target audience will
- able to describe digital repositories in general and DSpace implementation in particular
- understand the setup used at RNRA
- understand the important 'housekeeping' chores needed to maintain the health of the platform
- have knowledge and be comfortable in the use of the tools for managing and maintaining the platform

Content
This note is organized into the following topics:
1. Digital repositories
2. DSpace platform
3. DSpace software prerequisites
4. DSpace implementation at RNRA
5. Important management chores
6. Toolset for the technician
Digital Repositories

A digital repository is essentially a digital content management system (CMS) and at its core is the ability

- To allow a large number of people to share and contribute to stored data;
- Control access to data based on user role (i.e., define information users or user groups can view, edit, publish, etc.);
- Facilitate storage and retrieval of data;
- Control data validity and compliance;
- Reduce duplicate inputs;
- Simplify report writing;
- Improve communication among users.

A CMS is a tool that enables a variety of (centralised) technical and (de-centralised) non-technical staff to create, edit, manage and finally publish (in a number of formats) a variety of content (such as text, graphics, video, documents etc), whilst being constrained by a centralised set of rules, process and workflows that ensure coherent, validated electronic content.

There are a number of these CMSs and can be categorised by the task they are good at such as Web CMS like TYPO3, WordPress, Drupal, Joomla, etc; Digital Repositories; etc.

Digital repositories are increasingly being used to collect, archive, and disseminate all types of research outputs such as research articles, conference proceedings, dissertations, data sets, working papers and reports.

A Digital Repository such as DSpace preserves and enables easy and open access to all types of digital content including text, images, moving images, mpegs and data sets.
DSpace Platform

DSpace (http://www.dspace.org) is an open source software package that provides the tools for management of digital assets:

- **CAPTURES**
  - Digital material in any format
  - If desired, directly from creators (faculty, etc.)

- **DESCRIBES**
  - Metadata
    - Descriptive
    - Technical
    - Rights

- **DISTRIBUTES**
  - Via WWW, with necessary access control
  - For Example, http://www.repository.rnra.rw
  - Persistent identifiers “handles”
    - Persistent Identifier—globally unique—attached to objects
      - Communities
      - Collections
      - Items

- **PRESERVES**
  - Bitstream guaranteed

DSpace as a platform can be summarized as shown figure 1.
## Use Case Examples of DSpace

- **Institutional Repository** - Repository for staff, faculty and/or student papers, research, articles and publications at an institution. e.g. Deep Blue at University of Michigan ([http://deepblue.lib.umich.edu/](http://deepblue.lib.umich.edu/))

- **Image Repository** - Repository for digitized images of art or historical items to facilitate the use of the materials and preserve a record of the original items. e.g. Rice University’s Travelers in the Middle East Archive, TIMEA ([http://timea.rice.edu/](http://timea.rice.edu/))

- **Audio / Video Repository** - Repository for storing video files. e.g. Georgia Tech SMARTech ([https://smartech.gatech.edu/](https://smartech.gatech.edu/))

- **Museum / Cultural Heritage** - Preserve documents, publications, images and other items and provide access to the relevant community and beyond. e.g. New York University’s Afghanistan Digital Library ([http://afghanistandl.nyu.edu/](http://afghanistandl.nyu.edu/))

- **Government Records / Reports** - Repository for preserving and
distributing legal documents and records. e.g. World Health Organization Southeast Asian Region Digital Repository, India (http://repository.searo.who.int/)

• **Subject** - A repository focused on a specific area topic. e.g. The Belgian Poison Centre (http://www.poisoncentre.be/)

• **Learning Resources** - Repository for storing, managing, and facilitating the exchange of learning objects e.g. Ontario Council of University Libraries Cooperative Online Repository for Information Literacy (CORIL) (http://library.ryerson.ca/eresources/ozone/)

• **Federated Repositories / Networked Instances** - A collection of unique instances copied / networked to a central location e.g. Texas Digital Library (http://www.tdl.org/)

You can see more examples at http://www.dspace.org/use-cases

**DSpace Architecture**¹

The DSpace system is organized into three layers, each of which consists of a number of components as shown in the diagram.

The *storage layer* is responsible for physical storage of metadata and content. The *business logic layer* deals with managing the content of the archive, users of the archive (e-people), authorization, and workflow. The *application layer* contains components that communicate with the world outside of the individual DSpace installation, for example the Web user interface and the Open Archives Initiative protocol for metadata harvesting service.

Each layer only invokes the layer below it; the application layer may not use the storage layer directly, for example. Each component in the storage and business logic layers has a defined public API. The union of the APIs of those components are referred to as the Storage API (in the case of the storage layer) and the DSpace Public API (in the case of the business logic layer). These APIs are in-process Java classes, objects and methods.

¹ https://wiki.duraspace.org/display/DSDOC5x/Architecture
The data model can be seen at https://wiki.duraspace.org/display/DSDOC5x/Storage+Layer.
DSpace Software Prerequisites

These prerequisites are based on the requirements for DSpace 5.x

1. UNIX-like OS or Microsoft Windows
2. Oracle Java JDK 7 or OpenJDK 7
3. Apache Maven 3.0.5+ (Java build tool)
4. Apache Ant 1.8 or later (Java build tool)
5. Relational Database: (PostgreSQL or Oracle)
6. Servlet Engine (Apache Tomcat 7 or later, Jetty, Caucho Resin or equivalent)
7. Perl (only required for [dspace]/bin/dspace-info.pl)

DSpace can be installed and run on most modern PC, laptop or server hardware. However, if you intend to run DSpace for a large community of potential end users, carefully review the Hardware Recommendations).

DSpace Implementation at RNRA

The installation at RNRA is currently based on DSpace 5.3, however the stable version is now DSpace 5.4 released mid November 2015 which comes with some bug fixes. It is hoped that the current version installed at RNRA will be upgraded to the now stable version after launch. The upgrade will also provide us with the opportunity to add the responsive UI implemented in the Mirage 2 theme. Installing Mirage 2 on the current version presented us with some challenges.

The DSpace has been implemented to run inside the Tomcat7 Application Server on port 8080. In order to avoid users providing a port number 8080 when accessing it, an Apache Webserver has been installed as well which provides the access point for users on port 80. Although there are different ways of achieving this, proxing access through Apache webserver on port 80 is the recommended best practice. Proxing is achieved using AJP through mod_proxy. PostgreSQL is used for the Database Management System and XMLUI for the UI.

DSpace 5.3 is implemented with the following:

- Ubuntu Server 14.04 LTS
- Java JDK V7 (OpenJDK)
- Apache Maven 3.0.5 or later (Java build tool)
  - Maven is necessary in the first stage of the build process to assemble the installation package for your DSpace instance. It gives you the flexibility to customize DSpace using the existing Maven projects found in the [dspace-source]/dspace/modules directory or by adding in your own Maven project to build the installation package for DSpace, and apply any custom interface "overlay" changes.
- Apache Ant 1.8 or later
  - Apache Ant is required for the second stage of the build process. It is used once the installation package has been constructed in [dspace-
source]/dspace/target/dspace-installer and still uses some of the familiar ant build targets found in the 1.4.x build process.

- PostgreSQL 9.0 or later
- Apache Tomcat 7 or later
- Apache Webserver
  - This used as a proxy to the Tomcat using mod_proxy
- Perl 5
  - Only required for [dspace]/bin/dspace-info.pl

### Installation directories and files

The DSpace installation (content of this section is from DSpace Manual\(^3\)) consists of three separate directory trees:

- **The source directory**: This is where the source code lives. The directory is located at /usr/local/dspace-5.3-src-release.

- **The install directory**: This directory is populated during the install process and also by DSpace as it runs. It contains config files, command-line tools (and the libraries necessary to run them. The directory is located at /usr/local/dspace.

- **The web deployment directory**: This directory is generated by the web server the first time it finds a dspace.war file in its webapps directory. It contains the unpacked contents of dspace.war, i.e. the JSPs and java classes and libraries necessary to run DSpace. Files in this directory should never be edited directly; if you wish to modify your DSpace installation, you should edit files in the source directory and then rebuild. The contents of this directory aren’t listed here since its creation is completely automatic. The folder is located at /usr/local/dspace.

The installation of the DSpace was done in such a way that it will be ‘easy’ to manage and also upgrade. This means that the installation followed the recommended way of separating the core from those areas that are prone to customisations and this entailed the use of ‘overlays’. Overlays are best described in a slideshare presentation located at [http://www.slideshare.net/tdonohue/making-dspace-xmlui-your-own](http://www.slideshare.net/tdonohue/making-dspace-xmlui-your-own). Another considering made is the way Tomcat is configured to use DSpace webapps. The DSpace installation instructions describe either copying the DSpace webapps into the default Tomcat webapps folder or pointing Tomcat to the DSpace webapps by changing the appbase setting for localhost, we would not recommend either of these approaches – the latter would remove access to all the default Tomcat webapps, which may not be want you want. Instead we used the set Contexts for paths relevant to DSpace in Tomcat’s server.xml conf file.

---

\(^3\) https://wiki.duraspace.org/display/DSDOC5x/Directories+and+Files
As per DSpace standard, for the rest of the documentation, we will use [dspace-source] as the dspace source directory and [dspace] as the dspace install directory. [tomcat] is the tomcat7 directory located at /var/lib/tomcat7, [apache] is the apache2 directory located at /var/lib/apache2 and [log] is the directory where log files are found and is located at /var/log.

**Source Directory Layout**

- **[dspace-source]**
  - `LICENSE` - DSpace source code license.
  - `README` - Obligatory basic information file.
  - `build.properties` - The basic settings necessary to actually build/install DSpace for the first time
  - `dspace/` - Directory which contains all build and configuration information for DSpace
    - `bin/` - Some shell and Perl scripts for running DSpace command-line tasks. Primary among them is the 'dspace' commandline utility
    - `config/` - Configuration files:
      - `controlled-vocabularies/` - Fixed, limited vocabularies used in metadata entry
      - `crosswalks/` - Metadata crosswalks - property files or XSL stylesheets
      - `emails/` - Text and layout templates for emails sent out by the system.
      - `modules/` - Configurations for modules / individual features within DSpace
      - `registries/` - Initial contents of the bitstream format registry and Dublin Core element/qualifier registry. These are only used on initial system setup, after which they are maintained in the database.
      - `dspace.cfg` - The Main DSpace configuration file
      - `dc2mods.cfg` - Mappings from Dublin Core metadata to MODS for the METS export.
      - `default-license` - The default license that users must grant when submitting items.
      - `dstat.cfg`, `dstat.map` - Configuration for statistical reports.
      - `input-forms.xml`, `item-submission.xml` - Submission UI configuration files
      - `news-side.html` - Text of the front-page news in the sidebar, only used in JSPUI.
      - `news-top.html` - Text of the front-page news in the top box, only used in JSPUI.
      - `news-xmlui.xml` - Text of the front-page news, only used in XMLUI
    - `etc/` - This directory contains administrative files.
      - `postgres/` - Administrative scripts for PostgreSQL
      - `oracle/` - Administrative scripts for Oracle.
- **modules/** - The Web UI modules "overlay" directory. DSpace uses Maven to automatically look here for any customizations you wish to make to DSpace Web interfaces.
  - **jspui** - Contains all customizations for the JSP User Interface.
    - **src/main/resources/** - The overlay for JSPUI Resources. This is the location to place any custom `Messages.properties` files. (Previously this file had been stored at: `[dspace-source]/dspace/config/language-packs/Messages.properties`)
    - **src/main/webapp/** - The overlay for JSPUI Web Application. This is the location to place any custom JSPs to be used by DSpace.
  - **lni** - Contains all customizations for the Lightweight Network Interface.
  - **oai** - Contains all customizations for the OAI-PMH Interface.
  - **sword** - Contains all customizations for the SWORD (Simple Web-service Offering Repository Deposit) Interface.
  - **xmlui** - Contains all customizations for the XML User Interface (aka Manakin).
    - **src/main/webapp/** - The overlay for XMLUI Web Application. This is the location to place custom Themes or Configurations.
      - **i18n/** - The location to place a custom version of the XMLUI's `messages.xml`
      - **themes/** - The location to place custom Themes for the XMLUI. A theme for RNRA is located here and it is called `rnra`.

- **solr/** - Solr configuration files for all Solr indexes used by DSpace.
- **src/** - Maven configurations for DSpace System. This directory contains the Maven and Ant build files for DSpace.
- **target/** - (Only exists after building DSpace) This is the location Maven uses to build the DSpace installation package.
  - **dspace/dspace-installer** - The location of the DSpace Installation Package (which can then be installed by running ant update)

---

**Contents of XMLUI Web Application (aka Manakin)**
DSpace's Ant build file creates a dspace-xmlui-webapp/ directory with the following structure:

- (top level dir)
  - aspects/ - Contains overarching Aspect Generator config and Prototype DRI (Digital Repository Interface) document for Manakin.
  - i18n/ - Internationalization / Multilingual support. Contains the messages.xml English language pack by default.
  - themes/ - Contains all out-of-the-box Manakin themes
    - Classic/ - The classic theme, which makes the XMLUI look like classic DSpace
    - Kubrick/ - The Kubrick theme
    - Mirage/ - The Mirage theme (see Mirage Configuration and Customization)
    - Reference/ - The default reference theme for XMLUI
    - dri2xhtml/ - The base theme template, which converts XMLUI DRI (Digital Repository Interface) format into XHTML for display. See XMLUI Base Theme Templates (dri2xhtml) for more details.
    - dri2xhtml-alt/ - The alternative theme template (used by Mirage Theme), which also converts XMLUI DRI (Digital Repository Interface) format into XHTML for display. See XMLUI Base Theme Templates (dri2xhtml) for more details.
    - template/ - An empty theme template...useful as a starting point for your own custom theme(s)
  - WEB-INF/
    - lib/ - All the third-party JARs and pre-compiled DSpace JARs needed to run XMLUI
    - classes/ - Any additional necessary class files
    - cocoon.xconf - XMLUI's Apache Cocoon configuration
    - logkit.xconf - XMLUI's Apache Cocoon Logging configuration
    - web.xml - XMLUI Web Application configuration and Servlet mappings

Log Files
The first source of potential confusion is the log files. Since DSpace uses a number of third-party tools, problems can occur in a variety of places. Below is a table listing the main log files used in a typical DSpace setup. The ordering of the list is roughly the recommended order for searching them for the details about a particular problem or error.

<table>
<thead>
<tr>
<th>Log File</th>
<th>What’s In It</th>
</tr>
</thead>
<tbody>
<tr>
<td>[dspace]/log/dspace.log.yyyy-mm-dd</td>
<td>Main DSpace log file. This is where the DSpace code writes a simple log of events and errors that occur within the DSpace code. You can control the verbosity of this by editing the</td>
</tr>
<tr>
<td>File Path</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>![dspace-source]/config/templates/log4j.properties file and then running &quot;ant init_configs&quot;.</td>
<td></td>
</tr>
<tr>
<td>![dspace]/log/cocoon.log.yyyyMMdd-mm-dd</td>
<td>Apache Cocoon log file for the XMLUI. This is where the DSpace XMLUI logs all of its events and errors.</td>
</tr>
<tr>
<td>![log]/tomcat7/catalina.out</td>
<td>This is where Tomcat's standard output is written. Many errors that occur within the Tomcat code are logged here. For example, if Tomcat can't find the DSpace code (dspace.jar), it would be logged in catalina.out.</td>
</tr>
<tr>
<td>![log]/tomcat7/hostname_log.yyyy-mm-dd.log</td>
<td>If you're running Tomcat stand-alone (without Apache), it logs some information and errors for specific Web applications to this log file. Hostname will be your host name (e.g. dspace.myu.edu) and yyyy-mm-dd will be the date.</td>
</tr>
<tr>
<td>![log]/apache2/apache_log.yyyy-mm-dd.log</td>
<td>If you're using Apache, Tomcat logs information about Web applications running through Apache (mod_webapp) in this log file (yyyy-mm-dd being the date.)</td>
</tr>
<tr>
<td>![log]/apache2/error_log</td>
<td>Apache logs to this file. If there is a problem with getting mod_webapp working, this is a good place to look for clues. Apache also writes to several other log files, though error_log tends to contain the most useful information for tracking down problems.</td>
</tr>
<tr>
<td>![dspace]/log/handleplug.log</td>
<td>The Handle server runs as a separate process from the DSpace Web UI (which runs under Tomcat's JVM). Due to a limitation of log4j's 'rolling file appenders', the DSpace code running in the Handle server's JVM must use a separate log file. The DSpace code that is run as part of a Handle resolution request writes log information to this file. You can control the verbosity of this by editing ![dspace-source]/config/templates/log4j-handle-plugin.properties.</td>
</tr>
<tr>
<td>![dspace]/log/handleserver.log</td>
<td>This is the log file for CNRI's Handle server code. If a problem occurs within the Handle server code, before DSpace's plug-in is invoked, this is where it may be logged.</td>
</tr>
<tr>
<td>![dspace]/handle-server/error.log</td>
<td>On the other hand, a problem with CNRI's Handle server code might be logged here.</td>
</tr>
<tr>
<td>![log]/postgresql/postgresql-9.3-xxxx.log</td>
<td>PostgreSQL also writes a log file. This one doesn't seem to have a default location, you probably had to specify it yourself at some point during installation. In general, this log file rarely contains pertinent information--PostgreSQL is pretty stable, you're more likely to encounter problems with connecting via JDBC, and these problems will be logged in dspace.log.</td>
</tr>
</tbody>
</table>
Important Management Chores

There are various management chores that need to be run regularly. Although these are managed by cron jobs, there will be occasions that you may need to run them in an adhoc manner. These chores covers:

- Sitemap generations
- OAI Import
- Managing Discovery Indexes
- Managing Statistics
- Managing Email Subscriptions
- Filter Media
- Managing Curation Queue
- Managing Bitstream Integrity
- Cleaning up the database

Please see the following cron jobs setup to understand what the tasks are and how to run them from the command line. You have to be logged in as kimsuser user and then login as root.

Please note that due to permissions issues, the following cron jobs are now running under the root user.

```bash
# GLOBAL VARIABLES
# Full path of your local DSpace Installation (e.g. /home/dspace or /dspace or similar)
# MAKE SURE TO CHANGE THIS VALUE!!!
DSPACE = /usr/local/dspace

# HOURLY TASKS (Recommended to be run multiple times per day, if possible)
# At a minimum these tasks should be run daily.
# Regenerate DSpace Sitemaps every 8 hours (12AM, 8AM, 4PM).
# SiteMaps ensure that your content is more findable in Google, Google Scholar, and other major search engines.
0 0,8,16 * * * $DSPACE/bin/dspace generate-sitemaps > /dev/null

# DAILY TASKS
# (Recommended to be run once per day. Feel free to tweak the scheduled times below.)
# Update the OAI-PMH index with the newest content (and re-optimize that index) at midnight every day
# NOTE: ONLY NECESSARY IF YOU ARE RUNNING OAI-PMH
# (This ensures new content is available via OAI-PMH and
```
ensures the OAI-PMH index is optimized for better performance)
0 0 * * * $DSPACE/bin/dspace oai import -o > /dev/null

# Clean and Update the Discovery indexes at midnight every day
# (This ensures that any deleted documents are cleaned from the Discovery search/browse index)
0 0 * * * $DSPACE/bin/dspace index-discovery > /dev/null

# Re-Optimize the Discovery indexes at 12:30 every day
# (This ensures that the Discovery Solr Index is re-optimized for better performance)
30 0 * * * $DSPACE/bin/dspace index-discovery -o > /dev/null

# run the index-authority script once a day at 12:45 to ensure the Solr Authority cache is up to date
45 0 * * * $DSPACE/bin/dspace index-authority > /dev/null

# Cleanup Web Spiders from DSpace Statistics Solr Index at 01:00 every day
# NOTE: ONLY NECESSARY IF YOU ARE RUNNING SOLR STATISTICS
# (This removes any known web spiders from your usage statistics)
0 1 * * * $DSPACE/bin/dspace stats-util -i

# Re-Optimize DSpace Statistics Solr Index at 01:30 every day
# NOTE: ONLY NECESSARY IF YOU ARE RUNNING SOLR STATISTICS
# (This ensures that the Statistics Solr Index is re-optimized for better performance)
30 1 * * * $DSPACE/bin/dspace stats-util -o

# Send out subscription e-mails at 02:00 every day
# (This sends an email to any users who have "subscribed" to a Collection, notifying them of newly added content.)
0 2 * * * $DSPACE/bin/dspace sub-daily

# Run the media filter at 03:00 every day.
# (This task ensures that thumbnails are generated for newly add images,
# and also ensures full text search is available for newly added PDF/Word/PPT/HTML documents)
0 3 * * * $DSPACE/bin/dspace filter-media

# Run any Curation Tasks queued from the Admin UI at 04:00 every day
# (Ensures that any curation task that an administrator "queued" from the Admin UI is executed
# asynchronously behind the scenes)
0 4 * * * $DSPACE/bin/dspace curate -q admin_ui

----------------
# WEEKLY TASKS
# (Recommended to be run once per week, but can be run more or
# less frequently, based on your local needs/policies)
----------------
# Run the checksum checker at 04:00 every Sunday
# By default it runs through every file (-l) and also prunes old results (-p)
# (This re-verifies the checksums of all files stored in DSpace. If any files have been changed/corrupted, checksums will differ.)
0 4 * * * $DSPACE/bin/dspace checker -l -p
# NOTE: LARGER SITES MAY WISH TO USE DIFFERENT OPTIONS. The above "-l" option tells DSpace to check *everything*.
# If your site is very large, you may need to only check a portion of your content per week. The below commented-out task # would instead check all the content it can within *one hour*. The next week it would start again where it left off.
# 0 4 * * 0 $DSPACE/bin/dspace checker -d 1h -p

# Mail the results of the checksum checker (see above) to the configured "mail.admin" at 05:00 every Sunday.
# (This ensures the system administrator is notified whether any checksums were found to be different.)
0 5 * * 0 $DSPACE/bin/dspace checker-emailer

#----------------
# MONTHLY TASKS
# (Recommended to be run once per month, but can be run more or less frequently, based on your local needs/policies)
#----------------
# Permanently delete any bitstreams flagged as "deleted" in DSpace, on the first of every month at 01:00
# (This ensures that any files which were deleted from DSpace are actually removed from your local filesystem.
# By default they are just marked as deleted, but are not removed from the filesystem.)
0 1 1 * * $DSPACE/bin/dspace cleanup > /dev/null

#----------------
# YEARLY TASKS (Recommended to be run once per year)
#----------------
# At 2:00AM every January 1, "shard" the DSpace Statistics Solr index.
# This ensures each year has its own Solr index, which improves performance.
# NOTE: ONLY NECESSARY IF YOU ARE RUNNING SOLR STATISTICS
# NOTE: This is scheduled here for 2:00AM so that it happens *after* the daily cleaning & re-optimization of this index.
0 2 1 1 * $DSPACE/bin/dspace stats-util -s

sudo su - postgres
crontab -e

For the database (and you have to be logged in as postgres user):
sudo su - postgres
Clean up the database nightly at 4.20am
20 4 * * * vaxuumdb --analyze dspace > /dev/null 2>&1

Others

- **System availability considerations**
  - Includes making sure the system is up and running, may need to either constantly checking manually or employing an automated monitoring tool.

- **System performance considerations**
  - You need to clean up old files, especially that each build keeps a backup of the old state in order to manage disk space
  - Defragging the disk
  - Monitoring memory usage and fine tuning for optimum performance

- **Security considerations**
  - Using your available tools and experience to mitigate against security breaches.

- **Business continuity considerations**
  - This is mostly to do with backup and restore and there are notes on this located on the DSpace wiki pages at https://wiki.duraspace.org
  - At a minimum you should backup the [dspace-source] folder, [dspace]/config, [dspace]/assetstore and also a backup of the dspace database.

- **System/data integrity considerations**
  - Make sure that you pay attention to the checksum checker logs to make sure the bitstreams are not corrupted.
Toolset for the Technician

SSH Access using PUTTY
- PuTTY is a free implementation of SSH and Telnet for Windows and Unix platforms, along with an xterm terminal emulator. It is written and maintained primarily by Simon Tatham.
- Details at [http://www.chiark.greenend.org.uk/~sgtatham/putty/](http://www.chiark.greenend.org.uk/~sgtatham/putty/)

WinSCP
- Free SFTP, SCP and FTP client for Windows
- Details at [https://winscp.net/eng/index.php](https://winscp.net/eng/index.php)

PgAdmin
- pgAdmin is the leading graphical Open Source management, development and administration tool for PostgreSQL, running on Windows, Linux, Solaris, etc.
- Details at [http://www.pgadmin.org/](http://www.pgadmin.org/)

Webmin
- A web-based interface for system administration for Unix.

Nano
- A text editor for Unix-like computing systems or operating environments using a command line interface.
- Details at [http://www.nano-editor.org/](http://www.nano-editor.org/)

XML, XSLT and CSS
- XML stands for EXtensible Markup Language and was designed to store and transport data. XML was designed to be both human- and machine-readable.
  - Quick guide at [http://www.w3schools.com/xml/](http://www.w3schools.com/xml/)
- XSL stands for EXtensible Stylesheet Language, and is a style sheet language for XML documents. XSLT stands for XSL Transformations. In this tutorial you will learn how to use XSLT to transform XML documents into other formats, like XHTML.
  - Quick guide at [http://www.w3schools.com/xsl/](http://www.w3schools.com/xsl/)
- CSS is a stylesheet language that describes the presentation of an HTML (or XML) document. CSS describes how elements must be rendered on screen, on paper, or in other media.
  - Quick guide at [http://www.w3schools.com/css/](http://www.w3schools.com/css/)

Web developer tools (found in web browsers)
- Used to test and debug web UIs.
- Can be invoked on most web browsers by hitting F12
- In Chrome through menu and then More tools->Developer tools or Ctrl+Shift+I or simply by right mouse click and then on the context menu select Inspect element.
Google webmaster services
It has tools that let webmasters:

- Submit and check a sitemap.
- Allow for Google Analytics (GA) to monitor your traffic for statistics purposes - you need to obtain GA UA id that you need to apply in the dspace.cfg for GA to be able to collect statistics.
- Check and set the crawl rate, and view statistics about when Googlebot accesses a particular site.
- Write and check a robots.txt file to help discover pages that are blocked in robots.txt accidentally.
- List internal and external pages that link to the site.
- Get a list of links which Googlebot had difficulty crawling, including the error that Googlebot received when accessing the URLs in question.
- See what keyword searches on Google led to the site being listed in the SERPs, and the click through rates of such listings. (Previously named 'Search Queries'; rebranded May 20, 2015 to 'Search Analytics' with extended filter possibilities for devices, search types and date periods).
- Set a preferred domain (e.g. prefer example.com over www.example.com or vice versa), which determines how the site URL is displayed in SERPs.
- Highlight to Google Search elements of structured data which are used to enrich search hit entries (released in December 2012 as Google Data Highlighter).
- Demote Sitelinks for certain search results.
- Receive notifications from Google for manual penalties.
- Provide access to an API to add, change and delete listings and list crawl errors.


---

4 https://en.wikipedia.org/wiki/Google_Search_Console
Please see the following for further information

- [https://wiki.duraspace.org/display/DSDOC5x/DSpace+5.x+Documentation](https://wiki.duraspace.org/display/DSDOC5x/DSpace+5.x+Documentation)
- [https://wiki.duraspace.org/display/DSpace/Troubleshoot+an+error](https://wiki.duraspace.org/display/DSpace/Troubleshoot+an+error)
- [https://wiki.duraspace.org/display/DSDOC5x/AIP+Backup+and+Restore](https://wiki.duraspace.org/display/DSDOC5x/AIP+Backup+and+Restore)