Directories and Files

1 Overview
2 Source Directory Layout
3 Installed Directory Layout
4 Contents of Server Web Application
5 Log Files
    5.1 log4j2.xml File.

Overview

A complete DSpace installation consists of three separate directory trees:

- **The source directory:**: This is where (surprise!) the source code lives. Note that the config files here are used only during the initial install process. After the install, config files should be changed in the install directory. It is referred to in this document as *[dspace-source]*.
- **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), and usually -- although not necessarily -- the contents of the DSpace archive (depending on how DSpace is configured). After the initial build and install, changes to config files should be made in this directory. It is referred to in this document as *[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. It is usually referred to in this document as *[tomcat]/webapps /dspace*.

Source Directory Layout

- *[dspace-source]*
    - *LICENSE* - DSpace source code license.
    - *README* - Obligatory basic information file.
    - *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:
            - local.cfg.EXAMPLE - an example "local.cfg" file, which can be used to store all your local configuration overrides. See Configuration Reference.
            - *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.
            - entities/ - Configuration files for Configurable Entities
            - 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.
            - spring/ - Spring XML configurations used by DSpace for various features.
            - *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.
            - *submission-forms.xml* , *item-submission.xml* - Submission UI configuration files
        - *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. See also Advanced Customisation
        - *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 your DSpace installation package.
            - *dspace-installer*- The location of the DSpace Installation Package (which can then be installed by running *ant update*)

- The Source Release contains the following additional directories :-
    - *dspace-api* - Java API source module (to build the dspace-api.jar)
    - *dspace-oai* - OAI-PMH source module (to build to dspace-oai.jar)
    - *dspace-rdf* - RDF source module (to build to dspace-rdf.jar)
    - *dspace-rest* - REST API v6 (deprecated) source module
    - dspace-server-webapp - Primary backend webapp which hosts the REST API, along with any other enabled modules (OAI, RDF, SWORD, etc).
    - *dspace-services* - Common Services module
    - *dspace-sword* - SWORD (Simple Web-service Offering Repository Deposit) deposit service source module
    - *dspace-swordv2* - SWORDv2 source module
    - *pom.xml* - DSpace Parent Project definition

Installed Directory Layout

Below is the basic layout of a DSpace installation using the default configuration. These paths can be configured if necessary.

- *[dspace]*

- *assetstore/* - assetstore files. This is where all the files uploaded into DSpace are stored by default. See Storage Layer.
- *bin/* - shell scripts for DSpace command-line tasks. Primary among them is the 'dspace' commandline utility
- *config/* - configuration, with sub-directories as above
- *etc/* - Administrative and database management files
- *exports/* - temporary storage for any export packages
- *handle-server/* - Handles server files and configuration
- *imports/* - temporary storage for any import packages
- *lib/* - JARs, including dspace-api.jar, containing the DSpace classes
- *log/* - Log files
- *reports/* - Reports generated by statistical report generator
- *solr/* - Solr search/browse indexes
- *triplestore/* - RDF triple store index files (when enabled)
- *upload/* - temporary directory used during file uploads etc.
- *webapps/* - location where DSpace installs all Web Applications

Contents of Server Web Application

DSpace's Ant build file creates a webapps/*server/* directory with the following structure:

- server/
  - index.html - Root page of the third party HAL Browser (used to browse/search REST API)
  - login.html - (Custom) Login page for HAL Browser (supporting DSpace authentication plugins)
  - js/ - Javascript overrides for HAL Browser (main HAL Browser code is brought in via Spring REST dependencies)

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 locations given are defaults, and might be different for your system depending on where you installed DSpace and the third-party tools. The ordering of the list is roughly the recommended order for searching them for the details about a particular problem or error.

Log File	What's In It
[dspace] /log /dspace. log.yyyy-mm-dd	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 *[dspace-source]/config/templates/log4j.properties* file and then running "*ant init_configs*".
[dspace] /log /handle-plugin.log	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*.
[dspace] /log /handle-server.log	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.
[tomcat] /logs /catalina. out	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*.
[tomcat] /logs /hostname _log.yyyy-mm-dd.txt	If you're running Tomcat stand-alone (without Apache), it logs some information and errors for specific Web applications to this log file. *hos tname* will be your host name (e.g. *dspace.myu.edu*) and *yyyy-mm-dd* will be the date.
[tomcat] /logs /apache_lo g.yyyy-mm-dd.txt	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.)
[apache] /error_log	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.
PostgreSQ L log	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*.

log4j2.xml File.

the file *[dspace]/config/log4j2.xml* controls how and where log files are created. There are three sets of configurations in that file, called A1, and A2. These are used to control the logs for DSpace (as a whole), and the checksum checker respectively. As implied by the name, this configuration use Log4j v2. For more information on syntax, see https://logging.apache.org/log4j/2.x/manual/configuration.html