Module: Technical Basics

Module overview:
This module provides a basic technical overview of the DSpace software. The module will describe the three tiered application architecture of DSpace and look at how this relates to the server architecture of DSpace. The module will then look at what aspects of a DSpace repository should be backed up for both disaster recovery and preservation. The module will conclude with a look at the role of the repository administrator and the technical staff in configuring, managing and maintaining the repository.

Module objectives:
By the end of this module you will:

1. Understand the DSpace application architecture
2. Understand the DSpace server architecture
3. Know what and when to back up within DSpace
4. Understand the role of the repository administrator and the technical staff in configuring, managing and maintaining the repository (this will be discussed later in the course)

Practical Exercises:
For the practical exercise, please refer to your sheet ‘Local instructions’ for details of the following:

- How to launch a web browser
- The URL of the DSpace JSP user interface
The Application Architecture

- The DSpace system is organised into three tiers which consist of a number of components

- Each layer only invokes the layer below it i.e. the application layer may not used the storage layer directly

The application architecture

The DSpace system is organised into three tiers which consist of a number of components

Each layer only invokes the layer below it i.e. the application layer may not used the storage layer directly
The Storage Layer

The storage layer is responsible for physical storage of metadata and content.

DSpace uses a relational database to store all information about the organization of content, metadata about the content, information about e-people and authorization, and the state of currently-running workflows.
The Business Logic Layer

The business logic layer deals with managing the content of the archive, users of the archive (e-people), authorization, and workflow.
The Application Layer

- 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.

- The DSpace Web UI is the largest and most-used component in the application layer. Two versions:
  1. JSPUI: Built on Java Servlet and JavaServer Page technology
  2. XMLUI (Manakin): Built on XML and Cocoon technology

The Application Layer

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.

The DSpace Web UI is the largest and most-used component in the application layer. There are two versions:

1. JSPUI: Built on Java Servlet and JavaServer Page technology
2. XMLUI (Manakin): Built on XML and Cocoon technology
The Server Architecture

These systems may reside on a single server or be hosted separately on dedicated servers.

**The Server Architecture**

The *user interface* is the visual front end to the DSpace software. It is viewed through a web browser such as Microsoft’s Internet Explorer or Mozilla Firefox. There are two separate front ends to DSpace, these being the Java Server Page (JSP) interface and the Manakin interface. For instructions on how to view these, please see the ‘local instructions’ manual.

The *web application server*, either Apache Tomcat or Jetty sits between the User Interface and the Disk/File Store and Database and serves the web pages requested by the user of the repository.

The *disk/file store* is where items placed in the repository reside.

The *database*, either Postgres or Oracle is all the information about the organization of content, metadata about the content, information about e-people and authorization, and the state of currently-running workflows is stored.
The Server Architecture

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]`.  

**Web Deployment Directory [tomcat]/webapps/dspace**

Contains the JSPs and Java classes and libraries necessary to run 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]
  - dspace/
    - build.xml
    - bin/
    - config/
      - controlled-vocabularies/
      - crosswalks/
      - email/
      - language packs/
      - registries/
      - templates/
    - docs/
    - etc/
      - oracle/
    - modules/
      - jspsui/
      - lmi
      - osid
      - sword
      - xmlui
    - src/
    - target/

- dspace/
  - Directory which contains all build and configuration information for DSpace
  - build.xml - The Build file for Ant -- used to perform a fresh_install, upgrade, or deploy new changes.
  - CHANGES - Detailed list of code changes between versions.
  - KNOWN_BUGS - Known bugs in the current version.
  - LICENSE - DSpace source code license.
  - README - Obligatory basic information file.
  - bin/ - Some shell and Perl scripts for running DSpace command-line tasks.
  - config/ - Configuration files:
    - controlled-vocabularies/ - Fixed, limited vocabularies used in metadata entry
    - crosswalks/ - Metadata crosswalks - property files or XSL stylesheets
    - dspace.cfg - The Main DSpace configuration file (You will need to edit this).
    - 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 - Submission UI metadata field configuration.
The DSpace Course – Technical Basics

- **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 the JSPUI.
- **emails/** - Text and layout templates for emails sent out by the system.
- **language-packs/** - Contains "dictionary files" -- Java properties files that contain user interface text in different languages.
- **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.
- **templates/** - Configuration files for libraries and external applications (e.g., Apache, Tomcat) are kept and edited here. They can refer to properties in the main DSpace configuration - have a look at a couple. When they’re updated, a command line tool fills out these files with appropriate values from dspace.cfg, and copies them to their appropriate location (hence "templates").
  - **docs/** - DSpace system documentation. The technical documentation for functionality, installation, configuration, etc.
  - **etc/** - Miscellaneous configuration need to install DSpace that isn’t really to do with system configuration - e.g. the PostgreSQL database schema, and a couple of configuration files that are used during the build process but not by the live system. Also contains the deployment descriptors (web.xml files) for the Web UI and OAI-PMH support .war files.
  - **oracle/** - Versions of the database schema and updater SQL 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.
    - **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
  - **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-[version].dir** - The location of the DSpace Installation Package (which can then be installed by running ant update)
Installed Directory Layout

- [dspace]
  - assetstore/
  - bin/
  - config/
  - handle-server/
  - history/
  - lib/
  - log/
  - reports/
  - search/
  - upload/
  - webapps/

- [dspace]
  - assetstore/ - asset store files
  - bin/ - shell and Perl scripts
  - config/ - configuration, with sub-directories as above
  - handle-server/ - Handles server files
  - history/ - stored history files (generally RDF/XML)
  - lib/ - JARs, including dspace.jar, containing the DSpace classes
  - log/ - Log files
  - reports/ - Reports generated by statistical report generator
  - search/ - Lucene search index files
  - upload/ - temporary directory used during file uploads etc.
  - webapps/ - location where DSpace installs all Web Applications
### Log Files

- **[dspace]/log/dspace.log**
  - Main DSpace log file
  - Provides logging of events and errors that occur within the DSpace Code
  - Verbosity controlled by editing the [dspace]/config/templates/log4j.properties file and then running [dspace]/bin/install-configs.

- **[tomcat]/logs/catalina.out**
  - Tomcat standard output is written here
  - E.g. If Tomcat can't find the DSpace.jar the error would be written to catalina.out

---

**[dspace]/log/dspace.log**

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]/config/templates/log4j.properties file and then running [dspace]/bin/install-configs.

**[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. hostname will be your host name (e.g. dspace.myu.edu) and yyyy-mm-dd will be the date.
[tomcat]/logs/apache_log.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.

[dspace]/log/handle-plug.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]/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.

[dspace]/handle-server/error.log

On the other hand, a problem with CNRI's Handle server code might be logged here.

PostgreSQL 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.
What to Backup?

- What to backup?
  - Asset Store
    - This is where the bitstream files are located
  - Database
    - This is where information about organization of content, metadata about the content, information about e-people and authorization, and the state of currently-running workflows is stored
  - Source Directory
    - This is where the DSpace source code is located
  - Installation Directory
    - This is where the files are located which are used by DSpace as it runs

What to Backup

Asset Store - This is where the bitstream files are located.

Database - This is where information about organization of content, metadata about the content, information about e-people and authorization, and the state of currently-running workflows is stored.

Source Directory - This is where the DSpace source code is located.

Installation Directory - This is where the files are located which are used by DSpace as it runs.
The Repository Manager & Technical Staff

- Repository managers generally will manage the repository via the DSpace user interface

- Technical staff will be required to configure, customise and manage many features of the repository via the back end

- Examples of features that require configuration through the back end will be discussed throughout the course
Practical exercise: Familiarization

- Open a terminal window and browse the DSpace structure and log files to familiarise yourself. The location of these can be found in the local instructions sheet.

Start DSpace

In this exercise you start DSpace

1. Launch a terminal window by clicking “Terminal” on the desktop.

2. Navigate to the address of your DSpace installation. This can be found on the ‘local instructions’ sheet.

3. Familiarize yourself with the DSpace structure and log directories.
Credits

- These notes have been produced by:
  - Stuart Lewis & Chris Yates
  - Repository Support Project
    - http://www.rsp.ac.uk/
  - Part of the RepositoryNet
  - Funded by JISC
    - http://www.jisc.ac.uk/