
UNIT 7 CREATING DIGITAL LIBRARIES USING DSPACE

Structure

- 7.0 Objectives
- 7.1 Introduction
- 7.2 Functional Features of Dspace
- 7.3 Installing Dspace on Windows
- 7.4 Working with Dspace
- 7.5 Summary
- 7.6 Answers to Self Check Exercises
- 7.7 Keywords
- 7.8 References and Further Reading

7.0 OBJECTIVES

After going through this Unit, you will be able to:

- Describe the functional features of DSpace;
- Install Windows version of Dspace; and
- Create digital library using DSpace.

7.1 INTRODUCTION

DSpace is open source software, a turnkey repository application used by more than 1000+ organisations and institutions worldwide to provide durable access to digital resources. In India more than 140 institutions are using DSpace for building digital repositories.

DSpace is a software platform that enables organisations to:

- capture and describe digital material using a submission workflow module, or a variety of programmatic ingest options.
- distribute an organisation's digital assets over the web through a search and retrieval system.
- preserve digital assets over the long term.

The DSpace project was initiated in July 2000 as part of the HP-MIT alliance. The project was given \$1.8 million USD by HP over two years to build a digital archive for MIT that would handle the 10,000 articles produced by MIT authors annually. DSpace has gone through several versions and the current stable release available is version 4.2.

The DSpace Foundation was formed in 2007 as a non-profit organisation to provide support to the growing community of institutions that use DSpace. The foundation's mission is to lead the collaborative development of open source software to enable permanent access to digital works.

The code for DSpace is kept within a source code control system (<http://dspace.svn.sourceforge.net/viewvc/dspace/>) that allows code to be added or modified over time, whilst maintaining a track of all changes and a note of why the change was made and who made it. The Control of the source code repository is delegated to a small group of ‘committers’ who have the ability to change the code and release new versions. The committers work with the wider community of DSpace users to fix bugs and improve the software with new features.

In this we will guide you through the process of installation of DSpace (on a window platform) and familiarise you with the process of using and building collection in Dspace.

The Unit has been adapted from the DSpace official documentation and the Courseware developed by Aberystwyth University. Both the documents are available under the terms of either the GNU General Public License (<http://www.gnu.org/licenses/gpl.html>) and the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), for distribution and modification. The documents used are listed in the References and Further Readings section for further reference and you may refer them for further details.

7.2 FUNCTIONAL FEATURES OF DSPACE

The digital content in DSpace is presented in an organised tree structure of Community and Collections. Individual items can be accessed either through browsing the tree structure or searching with the Java freeware search engine Lucene built within. Each item gets a metadata description together with files available for download.

Full-text search : DSpace can process uploaded text based contents for full-text searching. Users may search for specific keywords that only appear in the actual content and not in the provided description.

Navigation : Users in DSpace find their way to relevant content through:

- **Searching** for one or more keywords in metadata or extracted full-text
- **Faceted browsing** through any field provided in the item description.
- Through **external reference**, such as a Handle
- **Browse** is another important mechanism for discovery in DSpace, whereby the user views a particular index, such as the title index, and navigates around it in search of interesting items.

Supported file types : While DSpace is most known for hosting text based materials including scholarly communication and electronic theses and dissertations (ETDs), it can accommodate any type of uploaded file. Files uploaded on DSpace are referred to as “Bitstreams” as after ingestion, files in DSpace are stored on the file system as a stream of bits without the file extension.

Optimized for Google Indexing : For the Google Scholar indexing, DSpace has added specific metadata in the page head tags that facilitates indexing in Scholar. Popular DSpace repositories often generate over 60% of their visits from Google pages.

OpenURL Support

DSpace supports the OpenURL protocol through linking server software called SFX server. DSpace will display an OpenURL link on every item page, automatically using the Dublin Core metadata if SFX server is implemented.

Self Check Exercise

Note: i) Write your answers in the space given below.

ii) Check your answers with the answers given at the end of this Unit.

1) Enumerate functional features of DSpace.

.....

.....

.....

.....

.....

Metadata Management

DSpace holds three types of metadata about archived content:

- **Descriptive Metadata:** A qualified Dublin Core metadata schema loosely based on the Library Application Profile set of elements and qualifiers is provided by default. However, one can configure multiple schemas and select metadata fields from a mix of configured schemas to describe items.
- **Administrative Metadata:** This includes preservation metadata, provenance and authorization policy data.
- **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.

Choice Management and Authority Control

This is a configurable framework that lets you define plug-in classes to control the choice of values for a given DSpace metadata fields. It also lets you configure fields to include “authority” values along with the textual metadata value. The choice-control system includes a user interface in both the Configurable Submission UI and the Admin UI (edit Item pages) that assists the user in choosing metadata value.

Licensing

DSpace offers support for licenses on different levels:

- Collection and Community Licenses
- License granted by the submitter to the repository
- Creative Commons Support for DSpace Items

Persistent URLs and Identifiers

Researchers require a stable point of reference for their works. To help solve this problem, a core DSpace feature is the creation of a persistent identifier for every

item, collection and community stored in DSpace. To persist identifiers, DSpace requires a storage- and location- independent mechanism for creating and maintaining identifiers. DSpace uses the CNRI Handle System for creating these identifiers.

Similar to handles for DSpace items, bitstreams also have ‘Persistent’ identifiers. They are more volatile than Handles, since if the content is moved to a different server or organisation, they will no longer work (hence the quotes around ‘persistent’). However, they are more easily persisted than the simple URLs based on database primary key previously used. This means that external systems can more reliably refer to specific bitstreams stored in a DSpace instance.

Getting content into DSpace

Rather than being a single subsystem, ingesting is a process that spans several. Below is a simple illustration of the current ingesting process in DSpace.

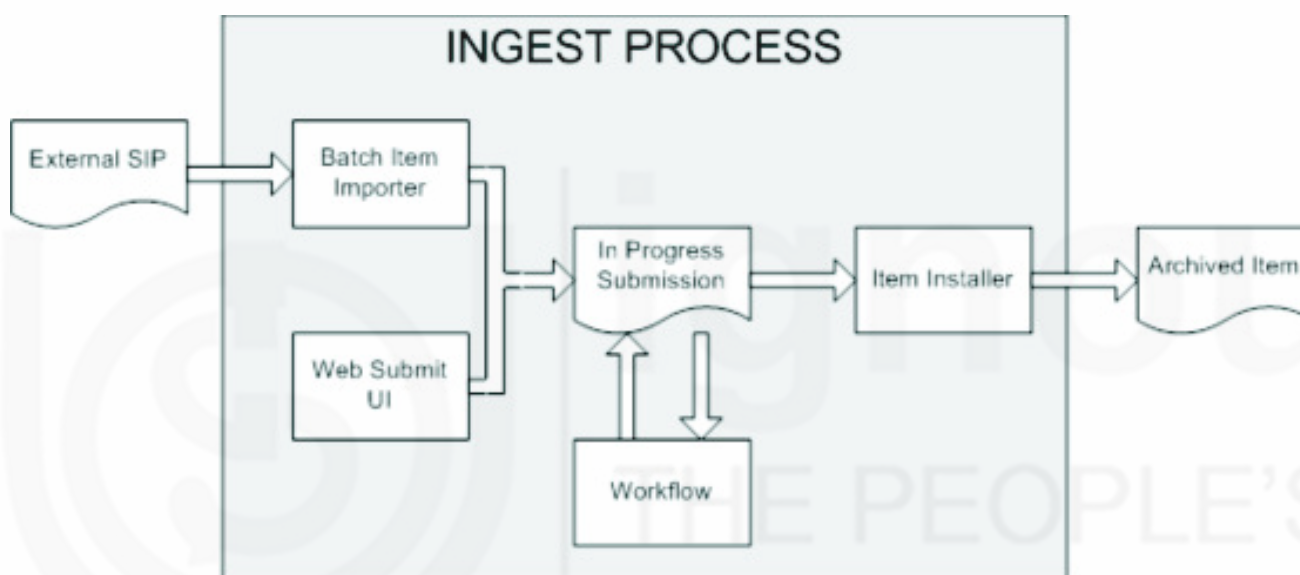


Fig. 7.1: DSpace Ingest Process

(Source: <https://wiki.duraspace.org/display/DSDOC4x/Functional+Overview>)

The batch item importer is an application, which turns an external SIP (an XML metadata document with some content files) into an “in progress submission” object. The Web submission UI is similarly used by an end-user to assemble an “in progress submission” object.

When the Batch Ingester or Web Submit UI completes the In Progress Submission object, and invokes the next stage of ingest (be that workflow or item installation), a provenance message is added to the Dublin Core which includes the filenames and checksums of the content of the submission. Likewise, each time a workflow changes state (e.g. a reviewer accepts the submission), a similar provenance statement is added. This allows us to track how the item has changed since a user submitted it.

Once any workflow process is successfully and positively completed, the In Progress Submission object is consumed by an “item installer”, that converts the In Progress Submission into a fully blown archived item in DSpace. The item installer:

- Assigns an accession date
- Adds a “date.available” value to the Dublin Core metadata record of the item

- Adds an issue date if none already present
- Adds a provenance message (including bitstream checksums)
- Assigns a Handle persistent identifier
- Adds the item to the target collection, and adds appropriate authorization policies
- Adds the new item to the search and browse index.

Workflow Steps

A collection’s workflow can have up to three steps. Each collection may have an associated e-person group for performing each step; if no group is associated with a certain step, that step is skipped. If a collection has no e-person groups associated with any step, submissions to that collection are installed straight into the main archive.

In other words, the sequence is this: The collection receives a submission. If the collection has a group assigned for workflow step 1, that step is invoked, and the group is notified. Otherwise, workflow step 1 is skipped. Likewise, workflow steps 2 and 3 are performed if and only if the collection has a group assigned to those steps.

When a step is invoked, the submission is put into the ‘task pool’ of the step’s associated group. One member of that group takes the task from the pool, and it is then removed from the task pool, to avoid the situation where several people in the group may be performing the same task without realizing it.

The member of the group who has taken the task from the pool may then perform one of three actions:

Workflow Step	Possible actions
1	Can accept submission for inclusion, or reject submission.
2	Can edit metadata provided by the user with the submission, but cannot change the submitted files. Can accept submission for inclusion, or reject submission.
3	Can edit metadata provided by the user with the submission, but cannot change the submitted files. Must then commit to archive; may not reject submission.

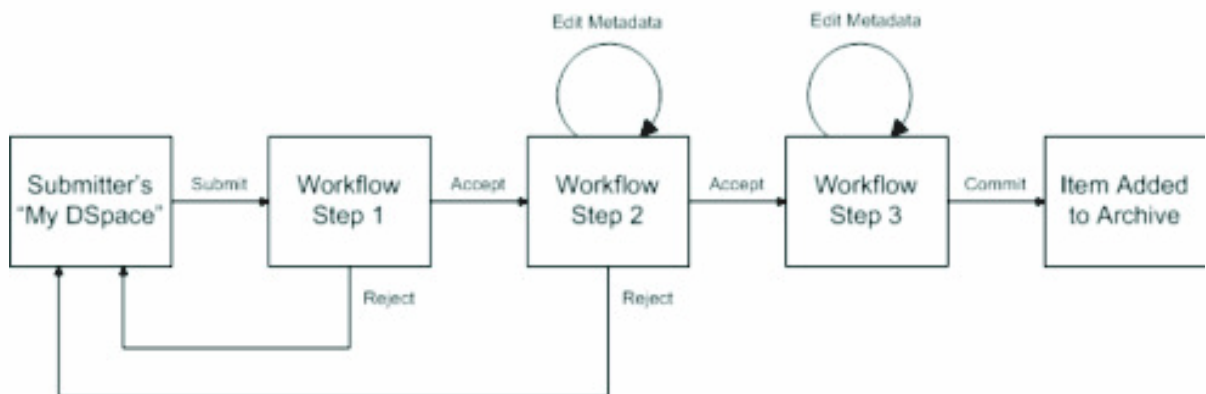


Fig. 7.2: Submission Workflow in DSpace

(Source: <https://wiki.duraspace.org/display/DSDOC4x/Functional+Overview>)

If a submission is rejected, the reason (entered by the workflow participant) is e-mailed to the submitter, and it is returned to the submitter's 'My DSpace' page. The submitter can then make any necessary modifications and re-submit, whereupon the process starts again.

If a submission is 'accepted', it is passed to the next step in the workflow. If there are no more workflow steps with associated groups, the submission is installed in the main archive.

One last possibility is that a workflow can be 'aborted' by a DSpace site administrator. This is accomplished using the administration UI.

The reason for this apparently arbitrary design is that it was the simplest case that covered the needs of the early adopter communities at MIT. The functionality of the workflow system will no doubt be extended in the future.

Command line import facilities

DSpace includes batch tools to import items in a simple directory structure, where the Dublin Core metadata is stored in an XML file. This may be used as the basis for moving content between DSpace and other systems.

Registration for externally hosted files

Registration is an alternate means of incorporating items, their metadata, and their bitstreams into DSpace by taking advantage of the bitstreams already being in accessible computer storage.

Getting content out of DSpace

- OAI Support

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.

- SWORD Support

SWORD (Simple Web-service Offering Repository Deposit) is a protocol that allows the remote deposit of items into repositories.

- Command Line Export Facilities

DSpace includes batch tools to export items in a simple directory structure, where the Dublin Core metadata is stored in an XML file.

- Packager Plugins

Packagers are software modules that translate between DSpace Item objects and a self-contained external representation, or "package". A *Package Ingestor* interprets, or *ingests*, the package and creates an Item. A *Package Disseminator* writes out the contents of an Item in the package format.

Crosswalk Plugins

Crosswalks are software modules that translate between DSpace object metadata and a specific external representation. An *Ingestion Crosswalk* interprets the external format and crosswalks it to DSpace's internal data structure, while a *Dissemination Crosswalk* does the opposite.

The Packager plugins and OAH-PMH server make use of crosswalk plugins.

Supervision and Collaboration

In order to facilitate, as a primary objective, the opportunity for thesis authors to be supervised in the preparation of their e-theses, a supervision order system exists to bind groups of other users (thesis supervisors) to an item in someone's pre-submission workspace. The bound group can have system policies associated with it that allow different levels of interaction with the student's item; a small set of default policy groups are provided:

- Full editorial control
- View item contents
- No policies

User Management

E-People and Groups are the way DSpace identifies application users for the purpose of granting privileges. Both E-People and Groups are granted privileges by the authorization system described below.

– User Accounts (E-Person)

DSpace holds the following information about each e-person:

- E-mail address.
- First and last names.
- Whether the user is able to log in to the system via the Web UI, and whether they must use an X509 certificate to do so.
- A password (encrypted), if appropriate.
- A list of collections for which the e-person wishes to be notified of new items.
- Whether the e-person 'self-registered' with the system; that is, whether the system created the e-person record automatically as a result of the end-user independently registering with the system, as opposed to the e-person record being generated from the institution's personnel database, for example.
- The network ID for the corresponding LDAP record, if LDAP authentication is used for this E-Person.

– Subscriptions

As noted above, end-users (e-people) may ‘subscribe’ to collections in order to be alerted when new items appear in those collections.

– Groups

Groups are another kind of entity that can be granted permissions in the authorization system. A group is usually an explicit list of E-People; anyone identified as one of those E-People also gains the privileges granted to the group.

Administrators can also use groups as “roles” to manage the granting of privileges more efficiently.

Access Control

Authentication

Authentication is when an application session positively identifies itself as belonging to an E-Person and/or Group.

Authorization

DSpace’s authorization system is based on associating actions with objects and the lists of EPeople who can perform them. The associations are called Resource Policies, and the lists of EPeople are called Groups. There are two built-in groups: ‘Administrators’, who can do anything in a site, and ‘Anonymous’, which is a list that contains all users. Assigning a policy for an action on an object to anonymous means giving everyone permission to do that action. The following actions are possible:

Usage Metrics

DSpace is equipped with SOLR based infrastructure to log and display page views and file downloads.

- Item, Collection and Community Usage Statistics

Usage statistics can be retrieved from individual item, collection and community pages.

- System Statistics

Various statistical reports about the contents and use of your system can be automatically generated by the system. These are generated by analyzing DSpace’s log files.

Digital Preservation

- Checksum Checker

The purpose of the checker is to verify that the content in a DSpace repository has not become corrupted or been tampered with.

System Design

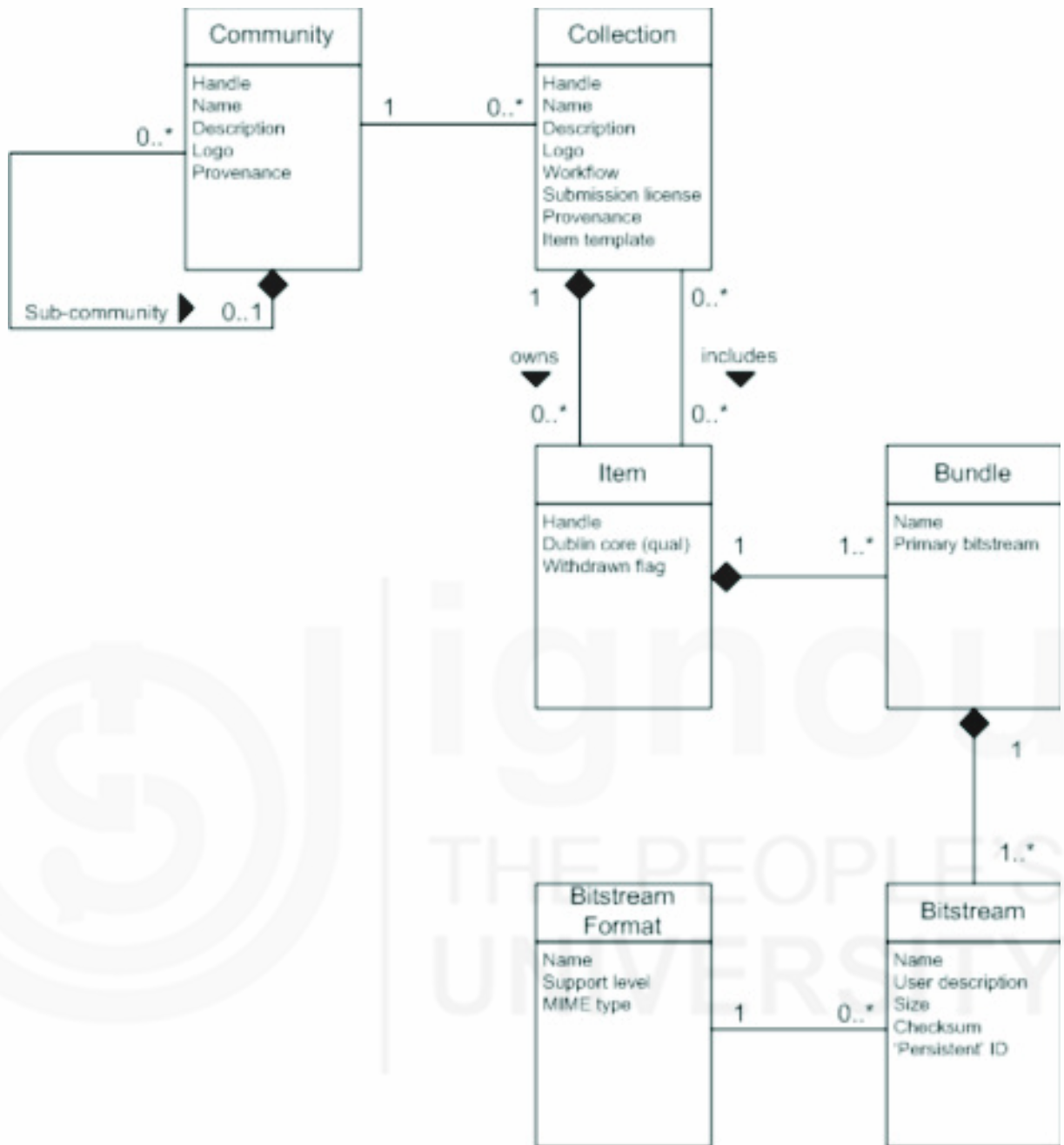


Fig. 7.3: Data Model Diagram

(Source: <https://wiki.duraspace.org/display/DSDOC4x/Functional+Overview>)

Each DSpace site is divided into *communities*, which can be further divided into *sub-communities* reflecting the typical university structure of college, department, research center, or laboratory.

Communities contain *collections*, which are groupings of related content. A collection may appear in more than one community.

Each collection is composed of *items*, which are the basic archival elements of the archive. Each item is owned by one collection. Additionally, an item may appear in additional collections; however every item has one and only one owning collection.

Items are further subdivided into named *bundles* of *bitstreams*. Bitstreams are, as the name suggests, streams of bits, usually ordinary computer files. Bitstreams that

are somehow closely related, for example HTML files and images that compose a single HTML document, are organized into bundles.

Storage Resource Broker (SRB) Support

DSpace offers two means for storing bitstreams. The first is in the file system on the server. The second is using SRB (Storage Resource Broker). Both are achieved using a simple, lightweight API.

SRB is purely an option but may be used in lieu of the server's file system or in addition to the file system. Without going into a full description, SRB is a very robust, sophisticated storage manager that offers essentially unlimited storage and straightforward means to replicate (in simple terms, backup) the content on other local or remote storage resources.

7.3 INSTALLING DSPACE ON WINDOWS

Running DSpace on Windows is actually rather similar to running it on any other operating system. For the most part, you should be able to follow the normal DSpace Installation Documentation. However, this page provides you with some hints that are specific to Windows.

Pre-requisite Software

You'll need to install this pre-requisite software (for DSpace 1.5.x and higher). Check the "Windows Installation" section of the System Documentation for the most recent pre-requisites, as they sometimes differ based on the version of DSpace you are running.

- **Java SDK (jdk-6u14-javafx-1_2-windows-i586)** : JDK is a development environment for building applications, applets, and components using the Java programming language. Download it from <http://java.sun.com/javase/downloads/widget/jdk6.jsp>. For Ant to work properly, you should ensure that JAVA_HOME is set.
- **PostgreSQL (8.x for Windows)** : PostgreSQL is a powerful, open source object-relational database system. It has native programming interfaces for C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, among others. This comes with a Windows installer app. Make sure the ODBC + JDBC options are selected, as well as the pgAdmin III tool. We will be using it for storing the database of our repository. You can download it from: <http://www.postgresql.org/download/windows>.
- **Apache Tomcat (apache-tomcat-5.5.28)** : An open source software implementation of the Java Servlets to serve as a Web server. You can download it from: <http://tomcat.apache.org/download-60.cgi>.
- **Apache Maven (2.2.1-bin)** : Apache Maven is a software project management and comprehension tool. Just unzip it wherever you want it installed, and add [path-to-apache-maven]\bin to your system PATH.
- **Apache Ant 1.7.x.** is a Java-based build tool. Just unzip it wherever you want it installed, and add [path-to-apache-ant]\bin to your system PATH.

Self Check Exercise

- Note:** i) Write your answers in the space given below.
 ii) Check your answers with the answers given at the end of this Unit.
- 2) What are the prerequisite software required for DSpace?

.....

General Installation Steps

- 1) Download the DSpace software from SourceForge (<http://sourceforge.net/projects/dspace/>) and other prerequisite software. Untar or unzip it and save it in a folder.
- 2) **Install JDK :** Double click and execute the installer file of Java that you have downloaded. Finish JDK installation by clicking Finish. Another installer will start automatically for installing **JRE**. Click next (or you may cancel it also)> Click finish to close the installer. Next is to set up the Environmental variables and JAVA HOME.

Right click on **My computer**> go to **Properties**> go to **Advanced TAB** > click on Environmental variables> select **PATH** in **system variables** section> click EDIT button. Open your program files directories in C drive and locate **JAVA** in Programme Files Directory> JAVA\JDK x.x.x.x\ bin (C:\Program Files\Java\jdk1.6.0_14\bin) folder. Copy the file path from the address bar of windows explorer. > Paste this path in system variable window (opened earlier) but use as a separator ‘semicolon’ (;) before it > Click ok. In **User Variable** segment> Click on **NEW** to set up a new user variable of **JAVA_HOME**. Variable name will : **JAVA HOME** and Variable Value: C:\Program Files\Java\jdk1.6.0_14 or according to your installed version. Give the path of your java home directory located in program files. Click ok, and apply the settings.

- 3) **Install Apache Maven and Apache Ant :** Extract the files of **Apache Maven and Apache Ant** into C drive. Then give path for apache maven in system variables: Right click my computer> properties>Advanced> Environmental variables > Click on path and edit it > Add path “C:\apache-maven-2.2.1 \bin” Now define path variable for apache ant in the same way Open the extracted folder of apache ant in C drive, copy the folder path from windows explorer address bar and paste it in system path. Click ok. This will complete the task of defining all system paths [**C:\Program Files\Java\jdk1.6.0_14\bin; C:\apache-maven-2.2.1 \bin; C:\apache-ant-1.8.0\bin**]. Now define **ANT HOME** in user variables. **Variable name: ANT HOME Variable value: C:\apache-ant-1.8.0** > Click ok and apply the settings. All system paths and user variables are defined. We can also check, what we have done till now. Open command prompt and run the following command to see the java version C:/> ‘java -version’ Same way

you can check ‘ant –version’ and ‘mvn –version’ and the command prompt will show relevant information regarding the respective software. If it appears all right then we may conclude that all packages java, maven and ant are successfully installed and paths are appropriately defined.

- 4) **Install Apache Tomcat :** Double click on Apache tomcat installer file and > Now, tick mark all the components in order to **do full installation** and then click next. > In this window give your **username and password**, that will give you access to monitor and control you tomcat server web interface. Then click next. > **Make sure that your java virtual machine path is appropriate with your JRE installation folder.** Click install. > Click finish... this will start tomcat service automatically. You will see Apache icon in Notification area of Taskbar.

- 5) **Install PostgreSQL :** Before installing PostgreSql check the file system of your local disc. It needs be NTFS. To identify this, right click the local disc > select “Properties”, see the “File system”. If all the drives in your system are FAT, then convert a convenient disc to NTFS. For converting, go to command prompt and type C:\>CONVERT C: /fs:ntfs this command will convert your c drive into NTFS file system. If you already have C drive with NTFS file system partition, you may simply proceed to install PostgreSQL by double clicking the installer file of postgresQL. You must provide the database password to administrate your DATABASE. Click next. > Check DATABASE port number. **The port number should be 5432.** After installation of Postgres SQL is over, the next task is to create database and login rolls. For this open pgAdminIII. Connect to the database (provide password). Database will start ... and then create login roll. > Right click on Login roles icon and click New Login Role. > Fill up the fields of role **name with dspace** and your **password is also dspace**. Then open role privileges tab. > Tick mark on icons named: Can create database objects, and can create roles. And then click ok. Login role is created. Now create Database. Right click on Database icon and click New Database. Fill up **Database name: dspace** and select **database owner dspace**. Click ok. Dspace database is created.

- 6) **Install DSpace :** Ensure the PostgreSQL service is running, and then run pgAdmin III (Start -> PostgreSQL 8.x -> pgAdmin III). Create the directory for the DSpace installation (e.g. C:\DSpace).

Build DSpace in the normal fashion. From [dspace-source]\dSPACE run:

```
mvn package
```

Then install DSpace to your specified location. From [dspace-source]\target\dSPACE-[version].dir run:

```
ant fresh_install
```

Create an administrator account, e.g. assuming C:\dSPACE is where your DSpace installation is:

```
C:\dSPACE\bin\dSPACE org.dSPACE.administer.CreateAdministrator
```

(then enter the required info)

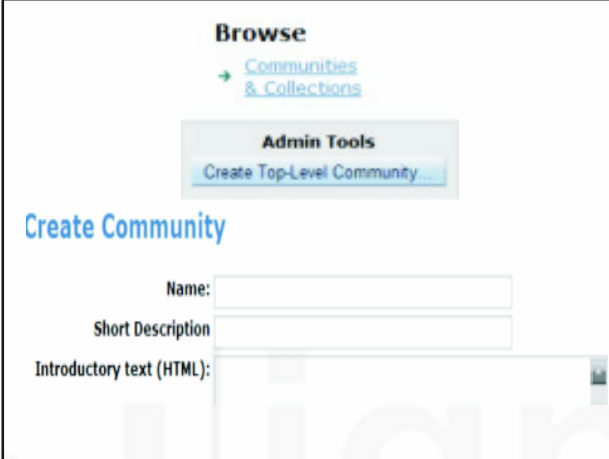
Copy the .war Web application files from C:\dSPACE\webapps to Tomcat's webapps dir, which should be somewhere like C:\Program Files\Apache Software Foundation\Tomcat\webapps

Start the Tomcat service


Browse http://localhost:8080/dspace. You should see the DSpace home page!

7.4 WORKING WITH DSPACE


1) Creating Communities

	<p>Sign in as an administrator</p> <ul style="list-style-type: none"> • Select 'Community & Collection' from the browse menu • Select 'Create Top-Level Community' from the Admin Tools menu • Complete the descriptive metadata for the Community • Click 'Create' to complete the Community
--	---

2) Creating Collection

	<p>Navigate to the parent Community of the collection to be created</p> <ul style="list-style-type: none"> • Select 'Create Collection' from the Admin Tools menu • Select the appropriate statements that apply to this collection
--	---

3) Descriptive Metadata for the Collection

	<ul style="list-style-type: none"> • Provide Descriptive Metadata for the collection • Select the users who can submit to the Collection and the 'Next' • Click 'Update' to complete the collection creation process
--	---

4) Creating a user and groups

Users require accounts to be able to log in and submit or edit items. Logical collections of users can be placed in groups to make administration easier.

DSpace has the facility User Self creation of account for which the following steps are to be followed:

- Click on My DSpace link
- Click on ‘New user? Click here to register.’
- Enter an email address and press ‘Register’
- Follow the link in the email that is sent for verification
- Provide name, telephone number, and a password
- New users have no privileges.

Users may be combined into logical groups for managing users and assigning privileges. Two special groups are possible on DSpace: i) Anonymous group in which there are no users in this group. Anyone can view the content without being logged, ii) Administrator group contains users who have full administrator access.

Administrator needs to be created directly on the DSpace server ([dspace]/bin/create-administrator) with the email address, first name, last name, and password details.

5) Metadata in DSpace

DSpace uses Dublin Core by default. Dublin core is made up of elements, and qualifiers. There are 15 base elements:

Title	Format
Creator	Identifier
Subject	Source
Description	Language
Publisher	Relation
Contributor	Coverage
Date	Rights
Type	

The elements can be further refined through the use of qualifiers as shown below in the case of the base DC element Title:

Schema = ‘dc’

Elements viz. Title / Creator / Subject / Description

Qualifiers e.g. Title.main / Title.subtitle / Title.series.

Multiple schemas can be held in the metadata registry of DSpace and the access for which is through Administer menu -> Metadata Registry.

Metadata Schema Registry

ID	Namespace	Name
1	http://dublincore.org/documents/dcmi-terms/	dc

Create a new schema by entering a namespace/name or edit an existing one by clicking the update button. The schema name must be less than 32 characters and cannot include spaces, periods or underscores.

Namespace:

Name:

A schema can be edited and submitted using the ‘Update’ button, deleted using the ‘Delete’ button next to an element and new elements can be added using the ‘Add Metadata Field’ section at the bottom of the page

64	title		Title statement/title proper.	<input type="button" value="Update"/>	<input type="button" value="Delete..."/>
143	type	qualificationlevel	Type Qualification Level Field	<input type="button" value="Update"/>	<input type="button" value="Delete..."/>
144	type	qualificationname	Type Qualification Name Field	<input type="button" value="Update"/>	<input type="button" value="Delete..."/>
66	type		Nature or genre of content.	<input type="button" value="Update"/>	<input type="button" value="Delete..."/>

Add Metadata Field

To create a new field you must provide a unique element and qualifier pair. The qualifier may be left blank if desired and the element and qualifier cannot contain spaces, underscores or periods.

Element:

Qualifier:

Scope Note:

6) Item submission Workflow

In the ‘Describe your Collection’ step while creating a new collection, one can select different workflow steps. During the process of creating the collection you will then be asked to select users and groups to assign to the workflow stages you have selected.

Describe the Collection

Please check the boxes next to the statements that apply to the collection. [More Help...](#)

<input checked="" type="checkbox"/>	New items should be publicly readable
<input checked="" type="checkbox"/>	Some users will be able to submit to this collection
<input type="checkbox"/>	The submission workflow will include an accept/reject step
<input type="checkbox"/>	The submission workflow will include an accept/reject/edit metadata step
<input type="checkbox"/>	The submission workflow will include an edit metadata step
<input type="checkbox"/>	This collection will have delegated collection administrators
<input type="checkbox"/>	New submissions will have some metadata already filled out with defaults

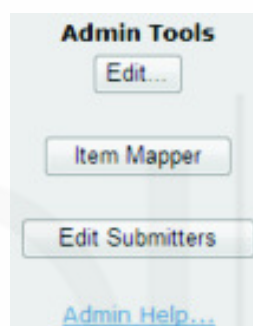
There are three options available for decision on the workflow:

- **Accept/Reject Step** – allows a user to simply accept an item, or reject it (with proper justification).
- **Accept/Reject/Edit Metadata Step** – allows a user to either accept or reject an item, and edit its metadata.
- **Edit Metadata Step** – allow the user to edit the metadata. This might be done to correct the metadata, or to improve it.

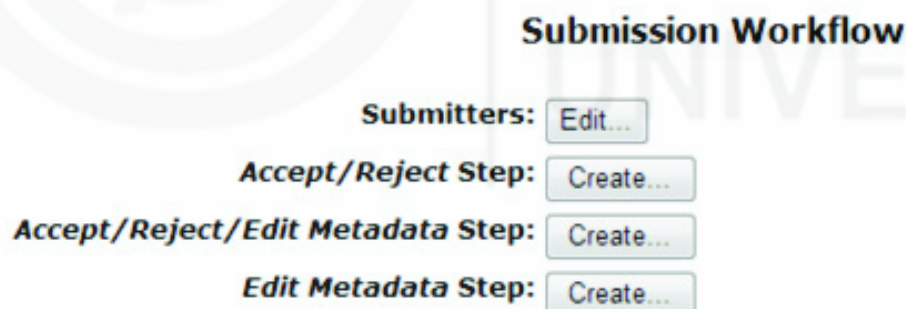
Any or all of the steps may be used. Workflow steps are worked through in order. If step 1 and 3 are selected, step 1 must be completed before step 3 will be initiated.

For an existing collection you may create the workflow through the following steps:

Log in as an administrator; go to the collection where you wish to create a workflow for. Click on the button 'Edit' in the 'Admin Tools' box.



Find the 'Submission Workflow' section, and click on whichever step you wish to create.



Edit the list of user and groups who can participate in the workflow as shown below:

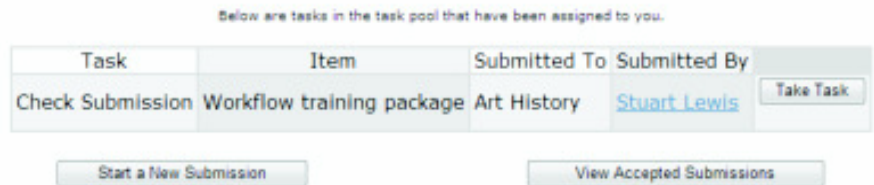


When you have finished, press ‘Update Group’.

Use the same process to edit and delete workflow in a collection.

Once an item has entered into a workflow, the concerned users and group members will receive an email alert that there is a task awaiting attention. When a user visits their ‘My DSpace’ page they will see any tasks in the pool.

Tasks in the Pool



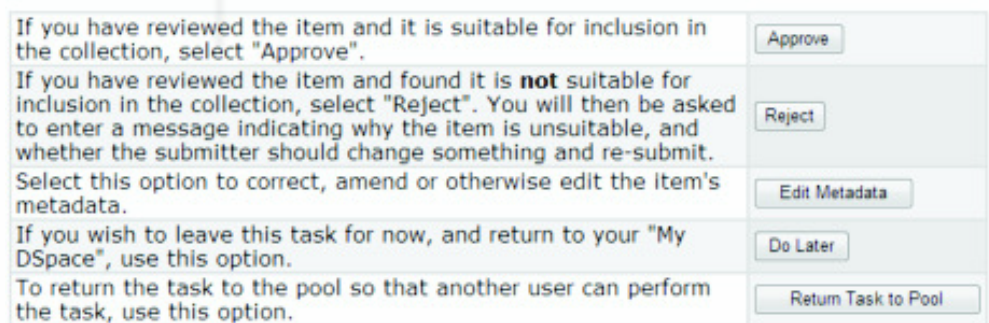
On clicking on ‘Take Task’ the user gets an overview of the item take a decision whether they wish to take the task.

Preview Task

The following item has been submitted to collection **Art History**. In order to accept the task of checking this item, please click "Accept This Task" below.



Clicking ‘Accept This Task’ will take the user into the workflow task page where they have several option for action such as, Approve, Reject, Edit Metadata, Do Later and Return Task to Pool.



7.5 SUMMARY

DSpace is a platform that allows you to capture items in any format – text, video, audio, and data and distribute it over the web. It indexes all the collection so that users can search and retrieve your items. It is best suited for preservation of digital work over the long term.

The Web-based interface of DSpace makes it easy for a submitter to create an archival item by depositing files. Data files, also called bitstreams, are organized

together into related sets. Each bitstream has a technical format and other technical information. This technical information is kept with bitstreams to assist with preservation over time. An item in DSpace is an “archival atom” consisting of grouped, related content and associated descriptions (metadata). An item’s exposed metadata is indexed for browsing and searching. Items are organised into collections of logically-related material.

In this Unit we have discussed in detail the technical features of DSpace along with the process of installation on your system and also using it for developing digital library.

7.6 ANSWERS TO SELF CHECK EXERCISES

- 1) The functional features of DSpace are:
 - Full-text search
 - Navigation
 - Supported file types
 - Optimized for Google Indexing
 - OpenURL Support
- 2) The prerequisite applications required for installation of DSpace are:
 - Java SDK (jdk-6u14-javafx-1_2-windows-i586)
 - PostgreSQL (8.x for Windows)
 - Apache Tomcat (apache-tomcat-5.5.28)
 - Apache Maven (2.2.1-bin)
 - Apache Ant 1.7.x.

7.7 KEYWORDS

- Bitstream** : a stream of data in binary form.
- Checksum Checker** : A checksum is a count of the number of bits in a transmission unit that is included with the unit so that the receiver can check to see whether the same number of bits arrived.
- OpenURL** : A standardised format of Uniform Resource Locator(URL) intended to enable Internet users to more easily find a copy of a resource that they are allowed to access.

7.8 REFERENCES AND FURTHER READING

The DSpace Course < <http://cadair.aber.ac.uk/dspace/handle/2160/615>>

DSpace Documentation <<https://wiki.duraspace.org/display/DSDOC4x/DSpace+4.x+Documentation>>