# Regional e-Repository Workshop on DSpace software

27 - 30 July 2011


**Edited by**

Dr. A.R.D. Prasad & Dr. Devika P Madalli

Faculty Members
Documentation Research and Training Centre
Indian Statistical Institute, 8th Mile Mysore Road
Bangalore 560059,
India


***Organized by***

**National Science Foundation**
**Sri Lanka**

***Sponsored by***

**UNESCO, Bangkok**

UNESCO
United Nations
Educational, Scientific and
Cultural Organization

NATIONAL SCIENCE FOUNDATION

# Training Manual

## Message from the Director

Adequate access to scholarly literature including databases and scientific journals has been identified as a major constraint faced by researchers and scientists in developing countries. This constraint is more prevalent in the Asia Pacific region where the national governments cannot afford to invest sufficient funds on such resources.

However, it is observed that the scholarly literature output in the world is dramatically increasing especially in the Asia Pacific region. The inadequacies and constraints for access to scholarly literature sources faced by researchers in poorer countries, could be compensated to some extent if a suitable collaborative mechanism could be developed among the countries for sharing the rich scholarly literature output in the region.

The technological advances brought about by ICT has opened numerous cost effective channels for such collaborative mechanisms and the digital revolution has introduced many changes and challenges in the information field resulting in the evolution of libraries into digital libraries and e-repositories. Operation of a regional network of e-repositories offering access to literature either for free or at a concessional rate could be an ideal solution to tackle the problem. Such a system will support new thinking, enrich education, accelerate research, and increase innovation outputs which would subsequently promote industrialization in the region in a sustainable manner.

I am extremely happy that the staff of the National Science Library and Resource Centre of the National Science Foundation is organizing this workshop to empower the Library and Information Community in the Asia region with necessary knowledge and skills for developing their institutional/national digital collections towards building up a regional network of e-repositories with an enormous potential for sharing the collective literature output in the region.

I wish to extend my sincere thanks to the UNESCO for their sponsorship and I am grateful to Professor A R D Prasad and Dr. Devika Medalli of the Indian Statistical Institute for their valuable contribution in conducting this workshop on D-space open source digital library software.

Dr Sarath Abayawardana

Director, National Science Foundation

2011. 07. 27

# Contents

# Digital Libraries: A Brief Introduction

**Devika P. Madalli**
Documentation Research Training Centre
Indian Statistical Institute
Bangalore
*devika@drtc.isibang.ac.in*

## Introduction

Digital Libraries (DLs) is an increasingly popular research area that takes off from research in traditional information retrieval or database techniques and progresses into more complex systems for online information services. The major boost to the development of the digital libraries comes from the web technologies that enable instantaneous online access to repositories. Often the question arises if the digital libraries are not same as or similar to web resources. There is however, a marked difference from just trying to get information from the web and organized information services rendered through digital libraries. The situation may be likened to a customer trying to find information at a bookstore and at a library. Digital libraries encompass a whole range of information services related work such as organization of digital information, information retrieval, user interfaces, archiving and preservation, services and social issues, evaluation and applications to particular areas and a set of standards for interoperability and value-added services.

## Desirable Features

Digital libraries are managed collections with in-built service layers. From the very first digital library initiatives which were just providing digitized collections, digital libraries have come a long way and have opened up many research avenues. This trend may be attributed to the increasing expectations of users and administrators of the digital libraries. As the services and applications are growing, DL software are being developed and refurbished to meet the changing user needs. DL packages are sophisticated systems with not only the basic components for building the digital repositories but also with efficient user interfaces for navigation, browse, search facilities. The latest trends in user centric design also concentrate on highly customized service interfaces according to each user's needs and level and preferred methods of interaction. A study into the desirable features of DL software points to the following (8):

- Structured
- Accessible
- Searchable
- Extensible
- Massive
- Heterogeneous
- Persistent

DLs are generally influenced by many factors such as the nature of the collections, the users and services planned, available infrastructure and technology. The end user only interacts through the user interfaces hence for most part user related features are discussed. However, the different modules that together make up for the DL's operations need to be examined under the following heads:

Architectural design – Modular and Open
Backend Database – scalable, robust, data formats
Network capabilities – web-based and seamless operations, persistent Ids, security and authentication
Metadata and Interoperability – compatible with world standards such as Dublin Core and OAI- PMH
Search, Retrieval and Display system – Type of Indexing, Ontologies, etc
User interfaces and customization – Annotations, Subscription
Ingest – XML based import/export
Service Layer

**Digital Library Issues**

Decision on the tools to be used for digital libraries and training staff to handle those form only one part of the issues related to digital libraries. More crucial issues relate to non-technical category such policies regarding the collection, access to collection, information model, funding and sustenance, security and IPR issues. Digital library software do not directly deal with or dictate any of these issues. However, the software should have the provision to implement the policy.

**Non-Technical Issues**

DL Technology is fairly seasoned and also available. What is more a challenge is to build content. IPR issues are the main concern for any online repository. Ownership issues, pre and post print rights and distribution policies have to be formulated in accordance

with the IPR relating to each DL's purview. Explicit decision has to be made on licensing and access policies.

Each organization would have its structure and typically DL collections are organized accordingly. The blueprint of the structure or hierarchy of the collection has to be made and Dspace enables implementing the structure that is approved through Information Model that basically enables forming communities and collections within them. The collection hierarchies are shown in the latest version.

**Technical Issues**

Several technical issues arise with respect to digital libraries. The basic infrastructure in terms of hardware and software and the decision whether to adopt open source software or opt for commercial packages. It may not always be true that users of open source software will have little or no technical help and that commercial ones are always supported. While these points can be argued for and against, it is important to take the final decision in view of the DI's collections, patrons and services, Infrastructure and funds available. The main tech issues to be considered maybe enlisted as:

- Open source software Vs. Commercial
- Operating system
- Hardware and peripheral requirements
- Network Components
- Standards – data formats, metadata, network, access, interoperability, encoding

**Approaches to Building Digital Libraries**

The two main approaches to building collection are:

1. Digitization -Retro-conversion of non-digital resources to digital

Most projects dealing with old materials like manuscripts and rare collections, epics, seminars and conference in retrospect have undertaken digitization projects. Projects study features of the scanners – flat bed and overhead scanners, OCR systems and data formats and accordingly configure the digitization equipment. The choice of the system and workflow depends of the project.

2. Digitally born resources – involves inter-conversion to standard formats and storage

Digital library collections are further categorized based on Discipline, resource-type, service or user-type based.

**DSpace Digital Library**

DSpace is open source software for building and managing Digital repositories. Developed jointly by MIT Libraries and Hewlett-Packard (HP), it is freely available to research institutions as an open source system that can be customized and extended. DSpace is a digital institutional repository that captures, stores, indexes, preserves, and redistributes content in digital formats. Institutional Repository is a set of services that a research institution/ organization/ university offers to the members of its community for the management and dissemination of digital materials created by the institution and its community members. Typically, DSpace has been deployed for Institutional Repositories of publications, thesis and dissertations.

There are several groups working on extending its capabilities such as implementation of ontologies in search interface and for submission module, customization for management of electronic theses and dissertations and for localization and international of the package for the world languages.

DSpace is designed for ease-of-use, with a web-based user interface that can be customized. The DSpace system provides a way to manage research materials and publications in a professionally maintained repository to give them greater visibility and accessibility over time.

**Why DSpace digital library**

DSpace is an open source system with a robust database support for trouble free DL operations. The choice of the software for a digital library should be based upon the nature of collections, data formats, applications, user expectations and infrastructure. Institutional repositories mostly comprise of resources such as publications, theses and dissertations and presentations. Ideally, for such collections the system should have simple installation, configuration and management facilities. This is especially required as in many instances collection building and its management maybe decentralized. Dspace has the following features that has made it popular choice for building Dls:

- Dspace is an open source technology platform which can be customized and its capabilities can be extended
- Dspace is a service model for open access and/or digital archiving for perpetual access.

- Dspace is a platform to build an Institutional Repository* and the collections are searchable and retrievable by the Web.
- To make available institution-based scholarly material in digital formats.
- The collections will be open and interoperable.

**Working with DSpace**

After installation and configuration all other functions such as building the collection, its organization, submission, review process, access and retrieval can be managed at distributed locations over the networks. The administrator is firstly responsible for implementing the information model and organizing the DL into communities and collections. The administrator can also delegate certain administrative tasks to others. Further, tasks such as reviewing, metadata verification are assigned to members. The admin creates groups of users (for instance grouping according to the departments they belong to) and authenticates users who can submit to the collection. Users can register themselves as members. Members can subscribe to entire collections or sub-collections depending on their interests. Mail alerts are sent to the members of each collection whenever a resource is added to that collection. Members authenticated by admin as 'submitters' can submit resources to the collection. The submitters are required to furnish metadata, basically Dublin Core data, for the resource they are submitting to the DL. Resources with a multiple files (such as website) can also be submitted. The submitters are required to agree with the terms and conditions of licensing set forth by the DL, before their submission can be passed on for review process. License information for every resource is stored. Dspace supports many popular data formats and has the provision for registering new bitstream formats.

Dspace is compliant with OAI-PMH version 2.0 and metadata in Dspace digital libraries can be harvested. Further there are various ways of access permissions that can be given to the items in the collections there by even highly classified information can be part of the collection but be made visible to only the person who may be authenticated by policies to view them.

**Architecture and system requirements**

The DSpace system is organized into three layers, each of which consists of a number of components. (www.dspace.org/docs)

- The storage layer: 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: containing components that communicate with the networked world outside of the individual DSpace installation, for example the Web user interface and the modules for metadata harvesting service.

It is not enough to just use open source software as they are free. It is equally important to examine whether the technologies needed to support it are also open and based on open standards. DSpace was developed to be open source, and in such a way that institutions and organizations with minimal resources could use it. The system is designed to run on the UNIX platform, and comprises other open source middleware and tools, and programs written by the DSpace team. All original code is in the Java programming language. Other pieces of the technology stack include a relational database management system (PostgreSQL), a Web server (Apache) and Java servlet engine (Tomcat), Jena (an RDF toolkit from HP Labs), OAICat from OCLC, and several other useful libraries. All these leveraged components and libraries are also open source software. The system is available on SourceForge (www.sourceforget.net), linked from both the DSpace informational web site and the HP Labs site (2)

***Information Model and Collection Organization***

Generally DL collections are categorized according to institutional structure and functions. Before any implementation of the DL software the DL library managers have to make a clear plan of the collection structure as suited for the purpose of the organization. DSpace has a well- planned information model to implement the collection structure. The digital library is divided into *communities* at the highest level. The communities can correspond to the different departments and units of an institution or organization. Communities further may have sub- communities within them. As of DSpace version 1.2, these communities can be organized into a hierarchy. Communities contain *collections*, which in turn contain *items*. Items are the actual resources that are uploaded into the DLs. Each item may belong to one collection. Each item contains bitstreams that are the computer files that make up the DL resource.

***DSpace Search System***

The end user can browse, search and access the collections using the hierarchies and also the alphabetic bar menu. For searching the collection, Dspace uses Lucene Search Engine, which is a part of Apache Jakarta Project (1). Additionally research projects such as the Ontology Add-on ( http://dspace-dev.dsi.uminho.pt:8080/en/addon_ontology.jsp) provides Ontologies that enables context based querying. This works like subject based directory structures.

Lucene search engine has very powerful search features that encompass many search approaches of the end-user. It provides the basic 'exact term' or keyword search. In addition it allows fielded search akin the field level search of library databases. In Dspace, Dublin Core elements are used for the field names. Lucerne also facilitates Boolean search, range searches, term boosting and proximity searches. The interesting search facility lucene uses fuzzy logic that is based on the Levenstien's alogorithm (5) that can replace and match terms by similarity. This feature is especially useful in instances where we hear a term and guess it spellings and more so in the case of personal names.

**Data Formats**

Resources are stored as bitstreams in Dspace repositories. In DSpace, a bitstream format is a unique and consistent way to refer to a particular file format. Each bitstream is associated with one Bitstream *Format*. Dspace supports most of the popular file formats. However it adapts varied levels for different file types. Actually these are levels set by the host institution of the DL what data formats they may be able to support at what level. By and large the three levels are categorized as:

***Supported:*** The format is recognized and the host institution will be responsible to make it usable in future also

***Known:*** The format is recognized, and the hosting institution will promise to preserve the bitstream as-is, and allow it to be retrieved. The hosting institution will attempt to obtain enough information to enable the format to be upgraded to the 'supported' level.

***Unsupported:*** The format is unrecognized, but the hosting institution will undertake to preserve the bitstream as-is and allow it to be retrieved.

**Metadata**

DSpace users deal with/come across metadata in the following modules:

- Administration modules: Dublin core registry, administrative metadata- default values, mail alert to subscribers
- Submission modules: descriptive metadata
- Harvesting – OAI-PMH using the DC elements (unqualified)
- Search result display: brief and full metadata

**Metadata in DSpace is of three types --**

***Descriptive Metadata:*** Dublin core qualified set of elements is used. Each item has one Dublin Core metadata record. Basically Dublin core has about 15 elements and the total qualifies set brings it up to 65 elements (4).

***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. Basically it is implementation of the Metadata Encoding and Transmission Standard (METS). It is partially implemented in Dspace up to the version 1.2.1

***Metadata harvesting:*** Dspace is compliant with the OAI-PMH ver 2.0 (7) for exposing metadata. OAI- PMH allows repositories to expose an hierarchy of sets in which records may be placed. DSpace exposes collections as sets. Each collection has a corresponding OAI set and harvestors use a verb (OAI- command) ListSets, to discover the sets. Only the 15 basic Dublin Core elements is exposed at present.

**Persistent Identifiers**

One of the common concerns of Digital library patrons is that online resources are volatile and may change or simply disappear. The idea of persistent identifiers of DLs is that it would be possible to find and retrieve deposited items in future and thus gain trust of the patrons. In particular, it is considered crucial that citations to archived material, whether found in printed articles or online, remain valid for long periods. DSpace has implemented CNRI handles (5) as the persistent identifier associated with each item. The CNRI Handle System covers assignment, management, and resolution of these persistent identifiers (or "handles").

## Advanced Features

### Customisation of DSpace

Customization in the context of DSpace can be done in three ways. One is

- using *dspace.cfg*
- using other facilities like Manakin, Mirage, controlled vocabularies, email alerts etc. which do not require knowledge JAVA or JSP
- using facilities in the operating system (LDAP, Shibboleth)
- changing the Code using JAVA, JSP

This workshop proceeding cover first two types of customization and does not attempt to cover the third and fourth ones as they would require knowledge of operating system and programming. Owing of some constraints (time mainly), it is not planned to cover all the facilities (mostly minor ones) DSpace provides.

### XMLUI Screens

After the initial JSPUI interface, the DSpace community realized the necessity of providing a facility for enthusiasts to develop various themes. As the JSPUI customization is too limited, XML based UI was developed using Manakin. Using Mankin at least three themes are in official release viz. Kubric, and Classic and the Default. The most recent alternative to Manakin is Mirage, which claims easier theme development and better performance. This volume explains how to use the themes in DSpace.

### Customization of Input data fields

There has always been requirement for accommodating local fields. From the first launch, DSpace was offering tips to accommodate local elements in addition to the Qualified Dublin Core adopted by DSpace.

### OAI-PMH and Metadata Schema

The Dublin Core is the standard metadata format, which most of the harvesting services use to aggregate many repositories, so that a single stop search engine can be offered to search many repositories. For example, DRTC offers a metadata harvester SDL (Search Digital Libraries – http://drtc.isibang.ac.in/sdl ). Though the OAI-PMH modules of DSpace can expose metadata to harvester using standard Dublin Core, it has facilities to expose in more than one metadata schema like MODS, METS, DIDL, QDC.

**Harvesting metadata from other Repositories**

DRTC has been using PKP Harvester software to provide an aggregator service SDL which collects metadata from Library and Information Science related digital repositories and provides common search facility to all the repositories. However, using the latest version of DSpace one can do the same in a collection. However, DRTC wishes to treat harvesting other repositories as a distinct service. But if somebody wishes to do the harvesting task done by DSpace, we have provided notes on this.

**Standardization of Data using Authority Files**

Library and Information professionals have done pioneering work in evolving data standardization wherever it is possible and necessary. For example, we have authority files for personal names, corporate names (organizations/institutions), series name, subject authority (controlled vocabulary) etc. Standardization ensures data exchange and interoperability. This is normally provided in many Library Management Software (LMS). DSpace has been making attempts to include the same. Earlier versions of DSpace did provided controlled vocabulary feature, however the latest versions added Name authority accessing LoC and publisher name & journal title accessing Romeo/Sherpa site of JISC project. This should entice library professionals for making best use of this facility.

**Other customization facilities**

There are host of customization facilities, which have brief self-explanations in dspace.cfg file. Some facilities are beyond our infrastructure and deemed as out of scope. For example SRB, which is basically running DSpace on grid architecture based network and hardly we come across in India any grid. So also the facility for Open URL, which requires a subscription based commercial service, which normally only very rich organizations must be subscribing.

**Conclusion**

DSpace has become the most widely used software for digital repositories. The following statistics are taken from OpenDOAR, as on 9th February, 2011. Total: 1857 repositories.

- The usage of **repository software** worldwide is:
  DSpace (692 – 37%)
  EPrints (301 – 16 %)
  Digital Commons (81 – 4%) (Berkeley Electronic Press)
  OPUS (54 – 3%)
  Greenstone (25 – 1%)

- **Type** of Repositories
  - Institutional Repositories (1517 – 82%)
  - Disciplinary Repositories (219 – 12%)
- Repositories **by Discipline**
  - Multidisciplinary (1177 – 63%)
  - Science in General (112 – 6%)
  - Social Science in General (89 – 4%)
  - Health and Medicine (141 – 7%)
  - History and Archeology (114 – 6%)
  - Library and Information Science (65 – 3%)
  - Mathematics and Statistics – (43 – 2%)
  - Computers and IT (87 – 4%)
- **Content** of the Repositories
  - Journal Articles (1208 – 65%)
  - ETD (966 – 52%)
  - Conference papers (651 – 35%)
  - Multimedia (435 – 23%)
  - Learning objects (285 – 15%)

Digital libraries today encompass a variety of resources, patrons and accordingly varied applications. The software package that powers DLs should therefore take into consideration the popular expectations of DLs. Dspace is designed for institutional repositories and is built with the MIT's experience of building institutional repositories. The features are quite adequate and useful for building institutional repositories. Further, it uses world standards and open standards for the DL operations that enhance its tenacity to bear with the technology upgrades in future.

**References**


Apache Jakarta Project: Lucene. http://jakarta.apache.org/lucene/docs/queryparsersyntax.html

DSpace Federation site. www.dspace.org

DSpace Technical Documentation at http://dspace.org/technology/system-docs/

DCMI site http:// www.dublincore.org

Gilleland, Michael. Levenshtein Distance, in Three Flavors. http://www.merriampark.com/ld.htm

Handle System. http://hdl.handle.net

Open Archives Initiative. http://www.openarchives.org

Seales, Brent. An Introduction to Digital Libraries At
http://dmn.netlab.uky.edu/~seales/cs585/lectures/week01/pdf/01-An%20Introduction%20to%20Digital%20Libraries.pdf (browsed on 3/02/2005)
Dspace : Ontology Add-on
http://dspace-dev.dsi.uminho.pt:8080/en/addon_ontology.jsp

# DSpace Administration

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

DSpace administration involves a whole range of tasks the administrator has to perform for successful creation and maintenance of a digital repository. When we first configure a digital repository using DSpace, we begin with creating Communities and collections. While creating communities and collections, we also have to take a decision with regard to who or which group can submit digital items to each collection. In addition, we also have to take a decision on who or which group of members (E-people) are authorized to review, approve and modify metadata of submissions.

**Communities and Sub-communities**

In DSpace, a digital repository *is* organized in terms of communities and collections and optionally one can create sub-communities under each community and also any number of levels of sub-communities under each sub-community. In other words, communities and sub-communities can be arranged hierarchically. One can use 'Dewey Decimal Classification (DDC) system and any other classification system depending on how you would like to organize your digital repository. For example, a digital repository of a university can be like

Philosophy
  Logic
  Epistemology
  Metaphysics
  Indian Philosophy
    Heterodox system
      Lokayata
      Jainism
      Buddhism
    Orthodox system
      Nyaya
      Vaiseshika
Sociology
  ...

...
Economics
...
...

Under each community or sub-community one can have any number of collections. Collections can NOT be subdivided into sub-collections. Ultimately, it is to the collections, authors submit digital documents (items).

**Note:** The communities and collections need not be subject-wise as shown in the above example. It can be anything, depending on how you would like organize your repository. For example, a digital repository of a software company can have communities like –

Administration
Marketing
Projects

The above communities can have sub- communities or collections under each community.

**Collections:** Collections can be created under communities or sub-communities. One way of organizing collections is by type of digital documents, they hold. For example, one can have collections like

Published Articles
Pre-prints
Presentations
Theses
Movies
Photographs

But again, there is no hard and fast rule, that the collections should be named like the above. You can choose anything keeping in view, the way your users expect it to be.

It is strongly advised that the administrator of a digital repository spends considerable time on the design aspect of his digital repository, as it is not good idea to modify/ delete communities or collections once the Digital Library is launched.

**Items:**

Each collection in a DSpace Digital Repository is populated with items, also called digital objects or digital documents. An item may have one or more bitstreams. In other words, an

item can have a bundle of bitstreams (files). An author (submitter) can upload more than one file for single document. For example, if a submitter wants to upload a thesis having many chapters and each chapter is a PDF file, he can upload all the PDF files together to be considered as one digital item. One can also submit a set of HTML files and mention which should be considered as the primary file, so that when the digital item is opened by any end-user, the primary file will be displayed first. Later the user can navigate through hyperlinks in the HTML files. In these cases, each file is considered as a bitstream, and all the files together constitute a digital item.

**Bitstream:**

Bitstreams in DSpace are the files having digital content. The bitstreams can be plain text files, or html files, audio or video files. In fact, DSpace accepts many file formats like PDF, MS-WORD, PostScript, mpeg, jpeg, tiff, gif etc. as bitstreams.

**Admin Tasks:**

1) Create Users who will act as submitters, reviewers, metadata editors, approvers
2) Create Communities
3) Create Collections
4) Create Collection authorizations

**Optional Steps:**

1) Creating Sub-communities
2) Creation of Groups
3) Modifying *Dublin Core Registry* (Discouraged)
4) Adding additional file formats to *Bitstream Format Registry*
5) Modifying or Deleting Digital Items
6) Modifying email alerts

**How to login to Administrator**

To login as DSpace administrator, one should enter the URL followed by '/dspace-admin'

For example: http://drtc.isibang.ac.in/jspui/dspace-admin

a) The 'http' can be 'https' if the repository is configured over a Secure Socket Layer (SSL). Ex: https://drtc.isibang.ac.in/jspui/dspace-admin
b) Sometimes, you may have to use port number after the host name, if the repository is configured on Tomcat or Jboss running as a port other than 80. Ex: http://drtc.isibang.ac.in:8080/jspui (Where 8080 is the port number)

c) Sometimes, it might have been configured on SSL with a port number, in which case the URL may look like: https://drtc.isibang.ac.in:8443/jspui/dspace-admin

Once the repository page is open you will see the login screen. It is looks like the login screen of any member of the digital repository.

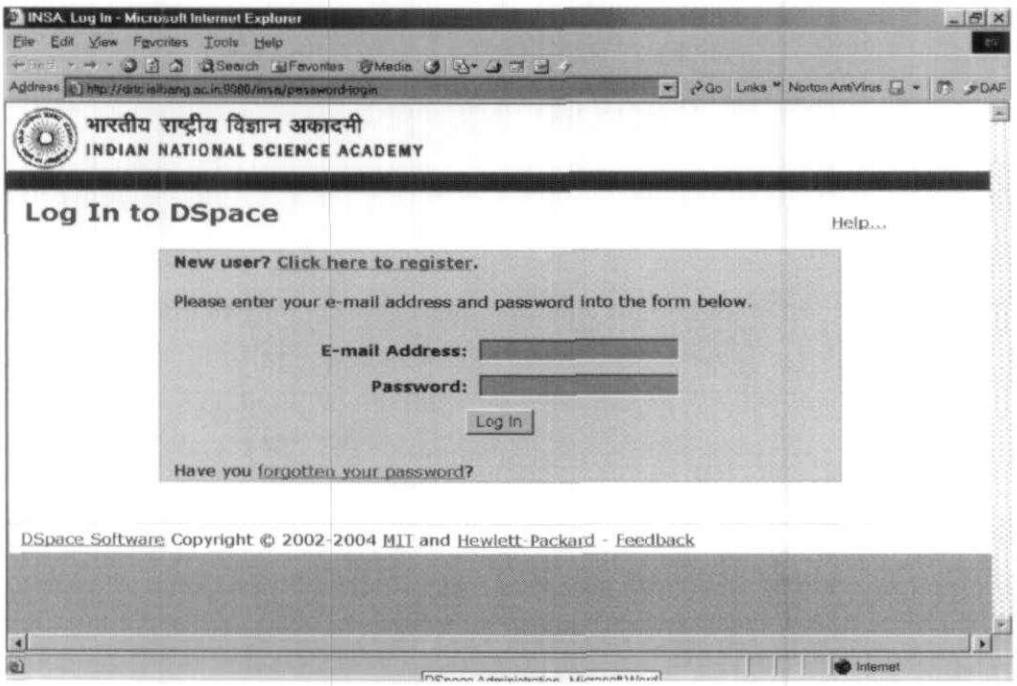


*Figure 1: Login Screen*

Here, you should enter your e-mail address and the password. DSpace will respond only if you are the Administrator. Otherwise, you will see a message stating 'Authorization Required' and 'You do not have permission to perform the action you just attempted'.

**Note:** The administrator account is created at the time of DSpace installation, using the following command

$DSPACE_HOME/bin/dspace create-administrator

Here, DSPACE_HOME is the directory where DSpace is installed, it could be '/dspace' or '/home/dspace' or any other directory.

After successful login, you should see the following screen, which offers a menu to perform various administrative tasks.

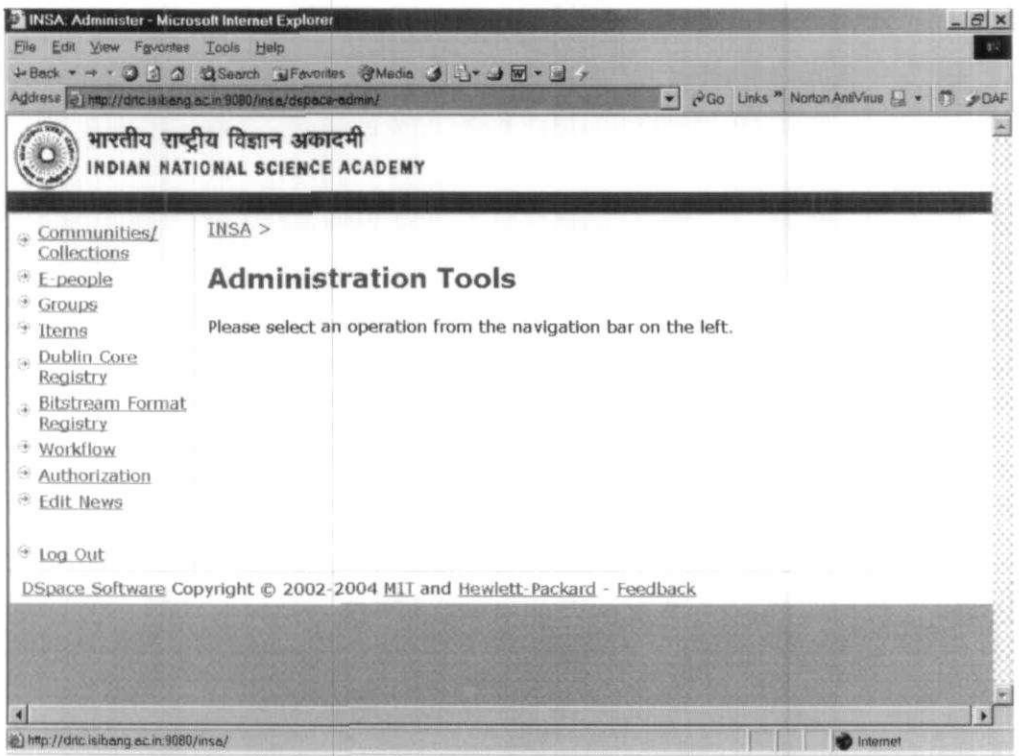


*Figure 2: Administration Tools*

**How to Create Initial Members**

Normally, anyone can become a member of a digital repository using 'My DSpace' in the left hand side menu of the first screen of the digital repository. However, it is desirable to have some initial members, who can take the responsibility of reviewing and approving. Though it is not mandatory to have members initially, to configure a digital repository for the first time, it is convenient to create some members or groups. When we first create communities and collections, the process can be completed by assigning who or which group is authorized to submit or review or modify metadata. Following are the steps the administrator should follow to add members to a digital repository.

Once the administrator has logged in, click 'E-People' on left hand side menu in the above figure 1. Then the following screen appears

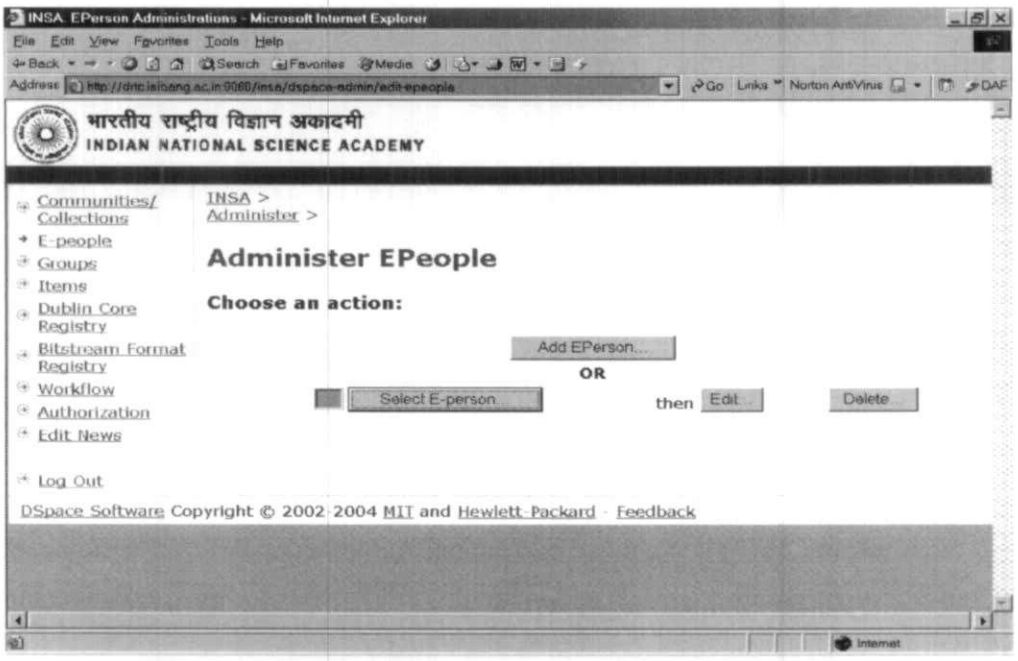


*Figure 3: Administer EPeople*

In the above screen, click 'Add EPerson' button, then you will get the following screen

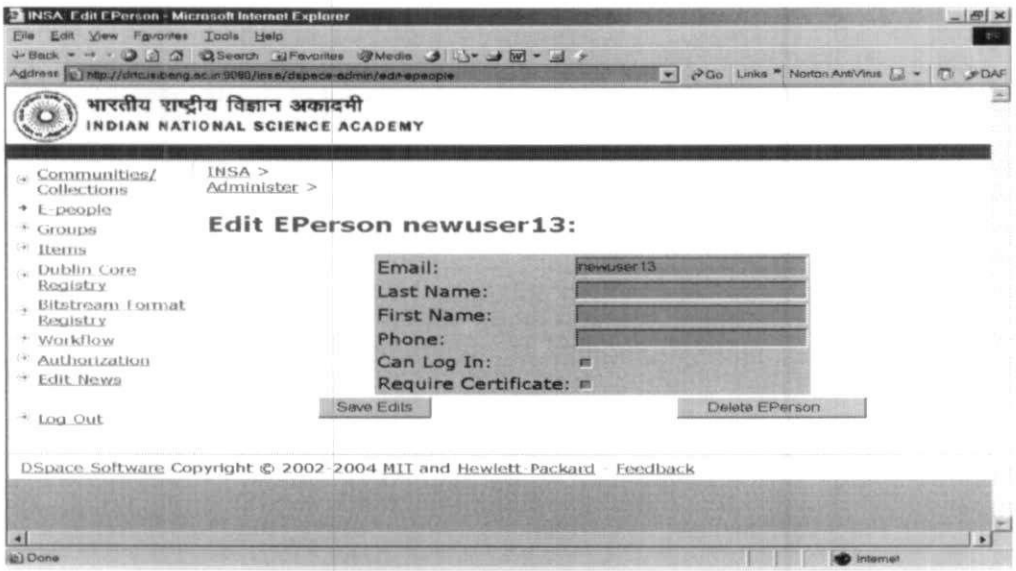


*Figure 4: Creating a new user*

In the above screen, enter the e-mail address, last name, first name etc, and click(tick) the square box opposite to 'Can Log In:' option. You can ignore 'Require Certificate:' option. Once, the details are filled in click 'Save Edits' button. This take you back to the earlier screen. You can verify, whether the system as added the new member by clicking, 'Select E-person'. Then the following screen appears and you can check whether the new member is added.


*Figure 5: E-People*

You can click 'Close' button to come out of the screen.

NOTE: Normally, an administrator need not use the facility of creating members. However, setting passwords for each member should be done by the member himself, using the 'Forgotten password' facility in the login screen (Figure 1).

**How to Create Communities, Sub-communities**

Firstly, you have to login as administrator, click 'administerr' on the left hand side menu. Once you see the screen with 'Administration Tools' (Figure 2), click 'Communities/ Collection'. Then the following screen appears

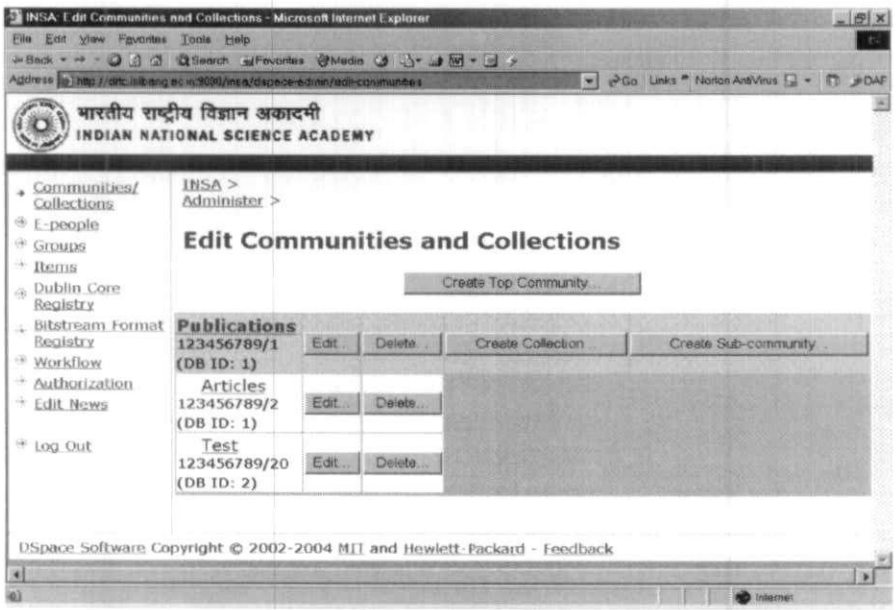


*Figure 6: Creation Communities and Collections*

To create a top level (first order in hierarchy of communities), you can click the button 'Create Top Community'. This is equivalent to creating communities like Philosophy, Sociology, Economics etc. in our example in the earlier section. If you wish to create sub-communities under a community, you should click 'Create Sub-community'. This is to create sub-communities like 'Logic', 'Epistemology', 'Metaphysics'. Creation of a community or sub-community involves providing

1) A name to the community (mandatory)

2) A short description of the community (optional)
3) An introductory text (in html format) about the community (optional)
4) A copyright note (optional)
5) Text (in html) to be appeared on the right column of the Digital repository page (optional)
6) A logo to appear on the community page (optional) and
7) A list of e-people, who can modify the logo (optional)

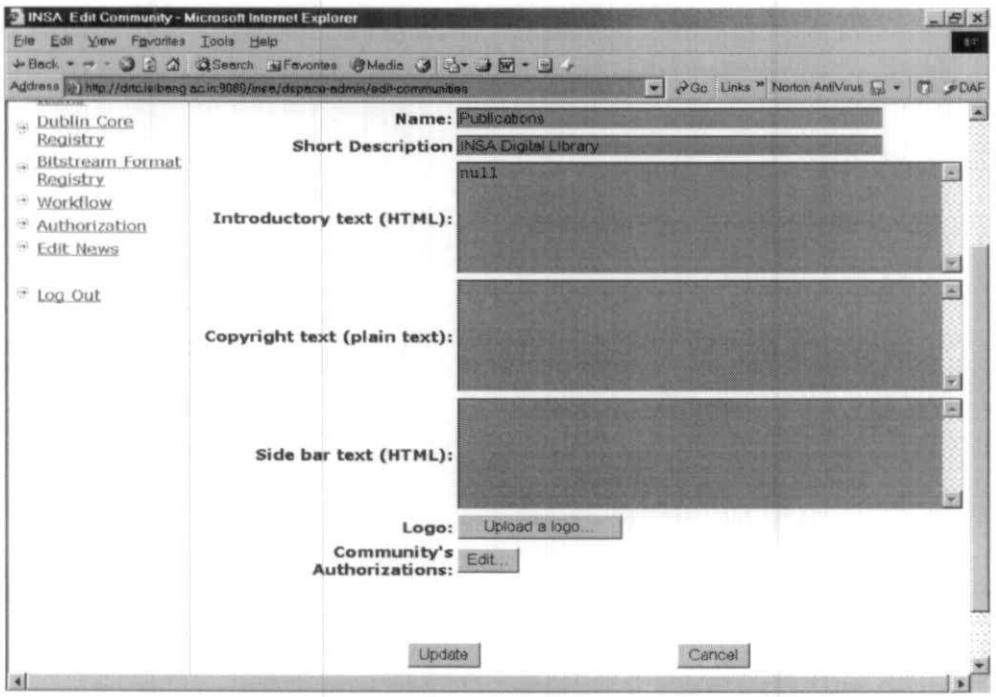


*Figure 7: Fields in Collection creation*

In the above screen

- Except the name of the community field, the rest of the fields are optional.
- You can enter a 'short description', which will appear on the community list page below the community name, and it should be one or two sentences of plain text describing the community.
- The 'introductory text,' 'side bar' and 'copyright text' fields are displayed on the community's home page. Please remember, the introductory text field and the side bar text field should be filled with appropriate html tags.

- To add a logo to be displayed on the community's home page, click on the **Upload Logo** button.
- On the next screen either type in a path to the logo file or browse to the logo file.
- Click on the **Upload** button.
- When all information for the Community Page is entered, click on the **Create** button.

**Note:** The Community will be given a default **READ** policy to **anonymous** group. Presently, since administration is done centrally, this tool doesn't have much use--usually you will just add **READ** permission for the **Anonymous** group just after you create a community, and leave it at that. This permission is applied to the community's logo if there is one.

In a similar fashion, sub-communities are created. When you 'create sub-community' the same screen appears, which allows you to describe a sub-community.

To add all the communities and sub-communities use the screen in Figure 6 repeatedly. Though the hierarchy of the communities and communities is not shown either in this screen (Figure 6), the top level communities appear on the digital repository page.

**How to create Collections**

Creating collections is essential. However, to create a collection you should first have created a community. Collections can appear under a community or sub-community. In other words, a community can have a combination of sub-communities and collections. Again under each sub-community there can be sub-sub-communities and/or collections. However, collection can not sub-divided into sub-collections.

It is the collections that hold digital items (also referred as digital documents, or digital objects).

To create a collection you should click 'Create Collection' in figure 6, which displays the following screen.

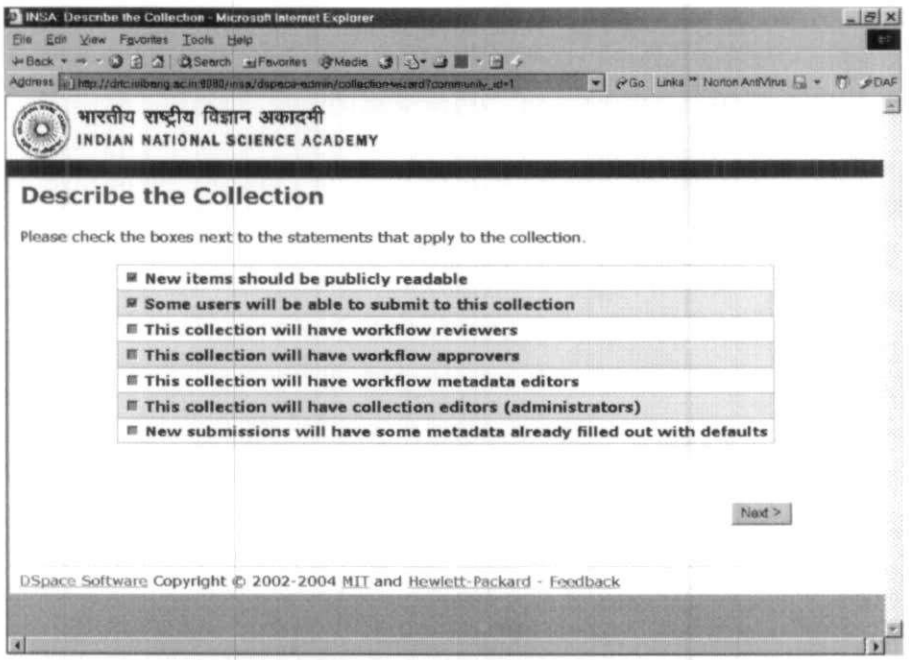


*Figure 8: Collection Policy*

By default, the first two, viz. 'New items should be publicly readable' and 'Some users will be able to submit to this collection', options are ticked. Before choosing the other options, the DSpace administrator should have clear idea of the workflow. DSpace allows you to create upto 3 workflows. It is not a good idea to avoid defining workflows. These workflows allow you to create a kind of moderation policy. If you do not define any workflow, the moment an author submits an item to the digital repository, the item will straight away will be deposited in the repository. In which case, there is every possibility of allowing authors to deposit irrelevant items. To filter out irrelevant documents, it is essential to define workflows. If the administrator defines all the workflows, DSpace sends e-mails to the reviewers of the collections, once a reviewer is satisfied with the submitted item and okays it, e-mails will be sent to metadata editors, once the metadata editors goes though the Dublin core elements and modifies them if necessary and okays the item. And finally the collection-administrator (if any) will approves the item. When the item is okayed by the approver, the item will be added to the digital repository. Following is the process an item goes through before it appears in the digital repository.

Submission, review (workflow 1) metadata validation (workflow 2) approval by collection administrator (workflow 3).

The DSpace administrator can decide on how many workflows are to be defined for each collection. For example, if he defines workflow-1 only, the item will be deposited in the repository immediately after a reviewer okays it. The collection policy can be set using the screen presented in Figure 8.

The following screens appear depending on the choices made in Figure 8. The figures 9a and 9b constitute one screen. If you scroll down the screen appearing 9a, you will see the screen in figure 9b. The fields are very much similar to the fileds that appear while creating a community and fairly self-explanatory.

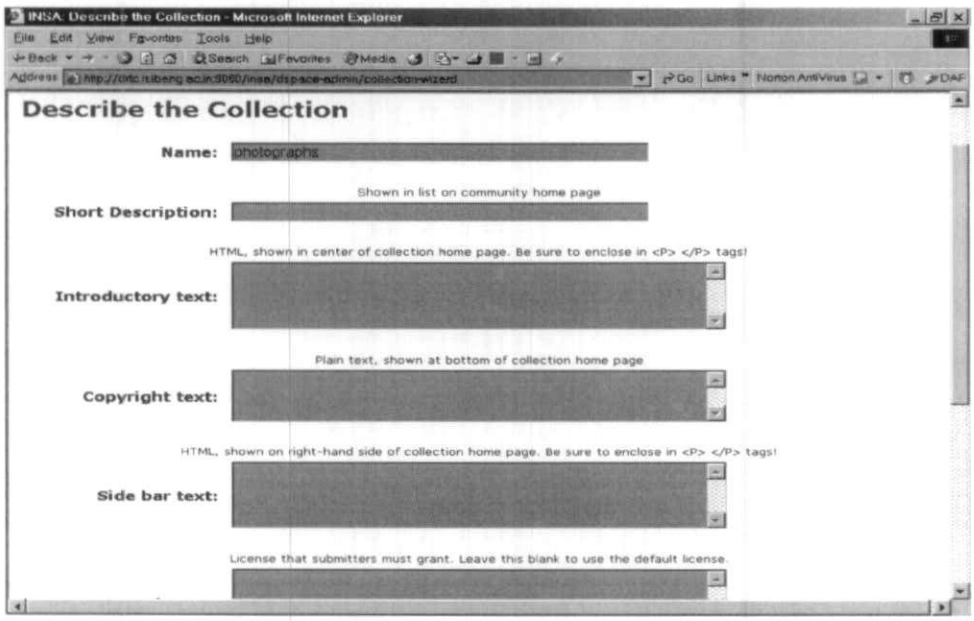


*Figure 9a: Collection Description*

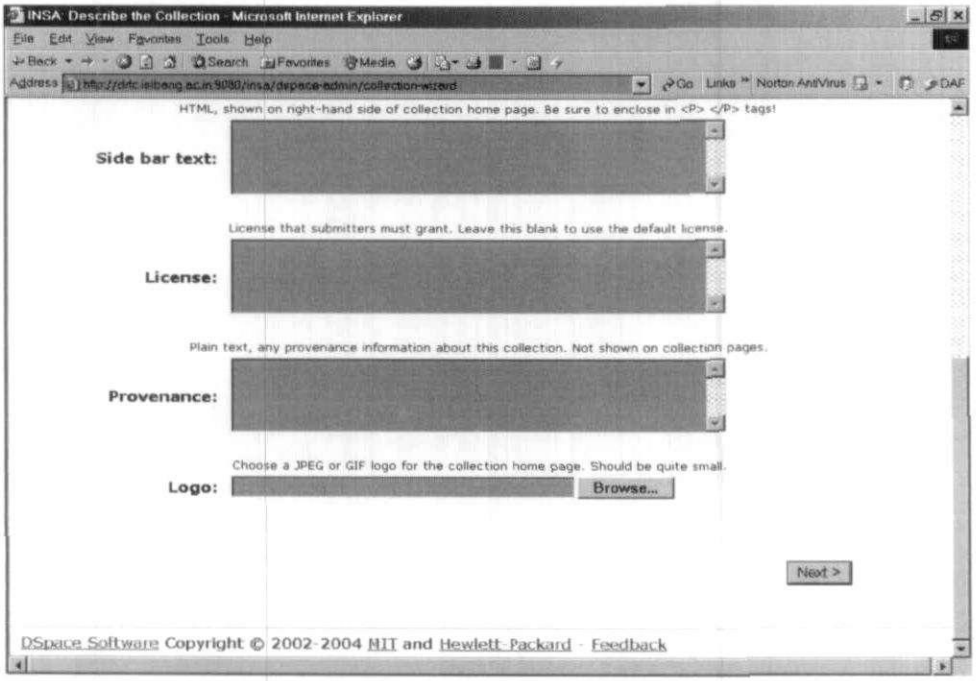


*Figure 9b: Collection Description*

The following screen (Figure 10), allows the administrator to add e-persons, who can submit to the collection. If you choose the button 'Select E-persons', the screen in the figure 5, will appear, where you can click the 'add' button to choose the list of e-persons who will be authorized to submit items to the collection. Normally, e-persons should write to the DSpace administrator his wish to submit to a collection. It should be noted, that an e-person who is authorized to submit to a particular collection, does not get authorization automatically to other collections in the digital repository. For example, if a user is authorized to submit to the thesis collection only, he may not be able to submit to 'pre-prints' collection, unless he is added to list of submitters of the 'pre-prints' collection'. The DSpace administrator can prevent any user submitting to a collection by removing his name from the list of submitters by selecting a user name and clicking the button 'remove selected' in figure 10.

The following screens appear one after other depending on the choices you made in the screen of Figure 8.

**Submitters:** Submitters are normally authors who are allowed to submit/ upload their publications to a digital repository. The submitters should already be members of the digital repository. Much of the DSpace organization is centered around the mail system. Hence, the membership is essentially providing the e-mail id of the members. It is the e-mail ids of the members that are added to various lists of submitters or reviewers or collection – administrators. Each collection will have separate list of submitters, reviewers, collection – administrators and metadata editors.

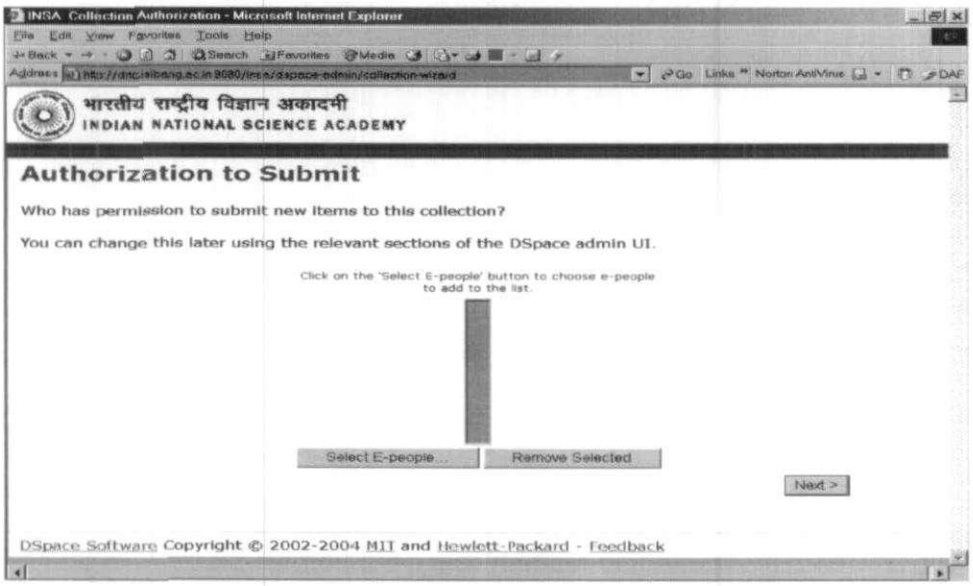


*Figure 10: Submission authorization*

In the above screen (Figure 10), if you click the button 'Select E-people', the screen in figure 5 will appear where you can select the e-people to be added to the list of submitters. If the administrator feels that a particular e-person has become problematic or does not wish to submit publications any more, the administrator can remove his name (e-mail id) from the list of submitters.

**Reviewers:**

Reviewers are members you can review the submissions to a digital repository. When an item is submitted to a collection, the DSpace generates an e-mail alert to the reviewers to facilitate the task of reviewing the item. A reviewer examines the item and can accept or reject the item, depending on the policy of the digital repository.

**Note:**

1) The normal practice is to have more than one reviewer, so that if one reviewer is away or busy the task of reviewing can be taken up any of the other members.
2) DSpace sends e-mail alerts to all the reviewers of a collection
3) Also when a reviewer logs in DSpace, he will be presented the list of submissions to be reviewed by him in a task pool
4) However, it does not mean that all the reviewers should accept the task of reviewing.
5) If one of the reviewers either accepts or rejects a submission, the task of reviewing the particular item will be automatically removed from the task list of other reviewers, though the e-mail alerts already sent can not be recalled!!!
6) A reviewer can not modify the metadata of an item
7) However, if the DSpace administrator has defined only one workflow, all the tasks of reviewing, metadata validation and approval can be bundled to the list of members in the workflow 1.

**Metadata Editors**

DSpace administrator can assign the task of validating the metadata of submitted items to a list of members. The metadata editors can examine the metadata and make any necessary modifications in case some of the values are wrongly entered. However, metadata editor/s can accept or reject a submitted item. Once the metadata editors okays the item, e-mail alert will be sent to the members in the list of collection – administrator

It is expected that the metadata editors are thorough with the Qualified Dublin Core and the way it is implemented in DSpace. The best practice is to have Librarians as metadata editors, whereas the reviewers should be subject experts of the discipline of the collection.

**Collection Administrators**

Collection administrators are like DSpace administrator, having all the power, except that their scope is limited to that particular collection only. He can not take administrative decisions on other collections of the repository. He can modify the list of submitters, reviewers, metadata editors, collection administratorss, change the description of the collection and even remove items from the collection. He can even modify the metadata at any time.

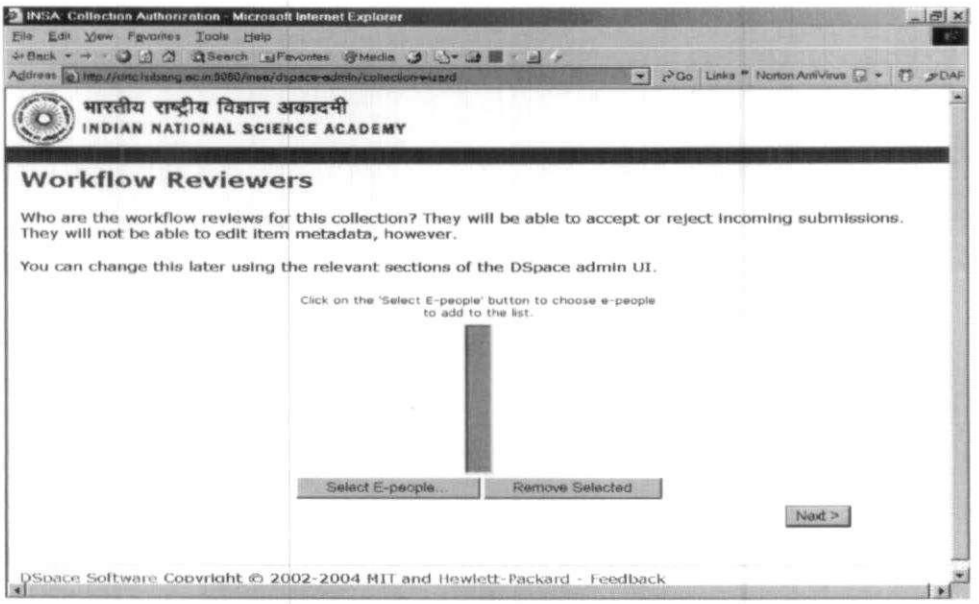


*Figure 11: Creating Reviewers*

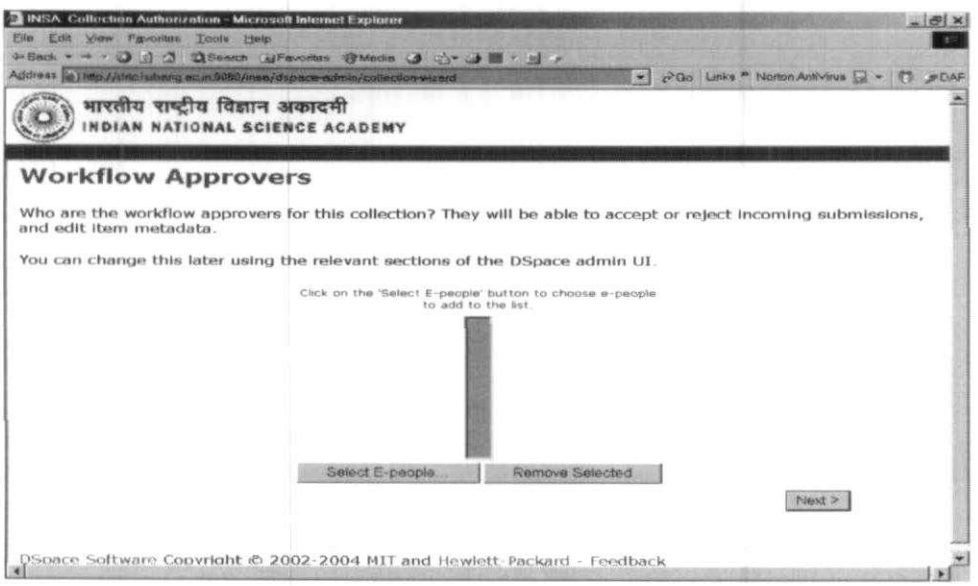


*Figure 12: Creating Approvers*

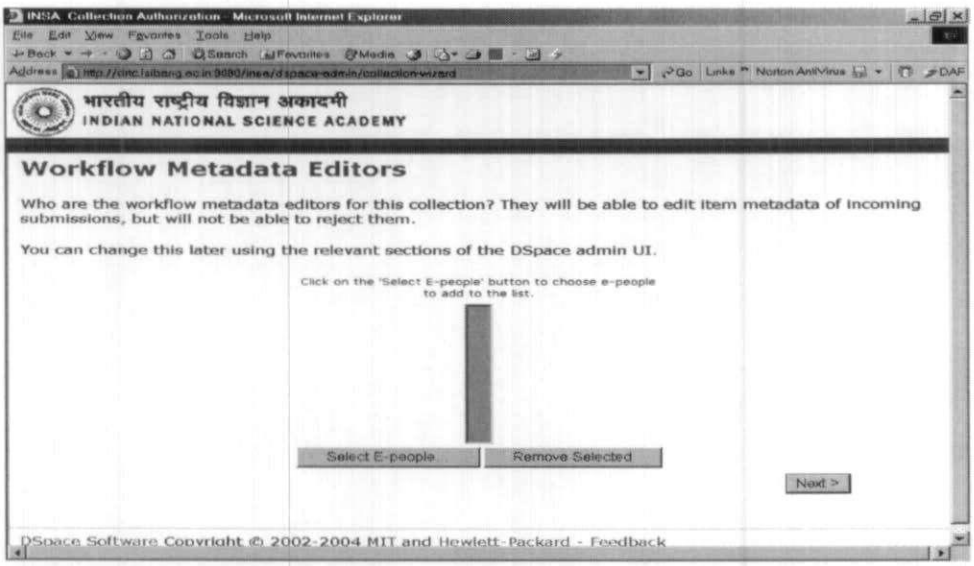


*Figure 13: Creating Metadata Editors*

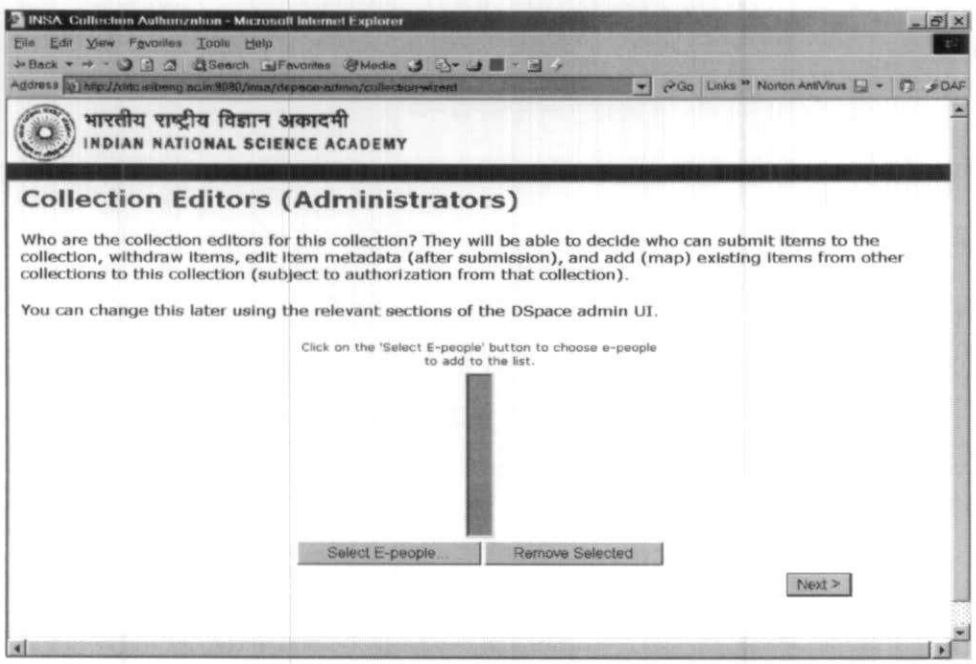


*Figure 14: Creating Collection Administrators*

**Default Metadata**

Either the DSpace administrator or the Collection administrator can set default values to some the fields in the Dublin core metadata. This facility is hardly used, as in many cases it is hard to have a constant value for any field in the Dublin core element set. Some of the elements that may be considered for setting up default values are:

1) Publisher (if it is repository of your institutes/ university) publications, where the publisher field always will have your organization name as the default value.
2) Type of digital item (if the collection is intended only for a particular type of items). For example, collection of 'ppts'.
3) Language (If the collection vastly consists of items in a particular language.
4) Sponsors (If majority of the items are sponsored by one sponsorer)

Note: These default values are presented to the author/ submitter while submitting an item to a repository. However, the author can always override the default values at the time of submission. It is just to save the time of the author while filling up certain fields.

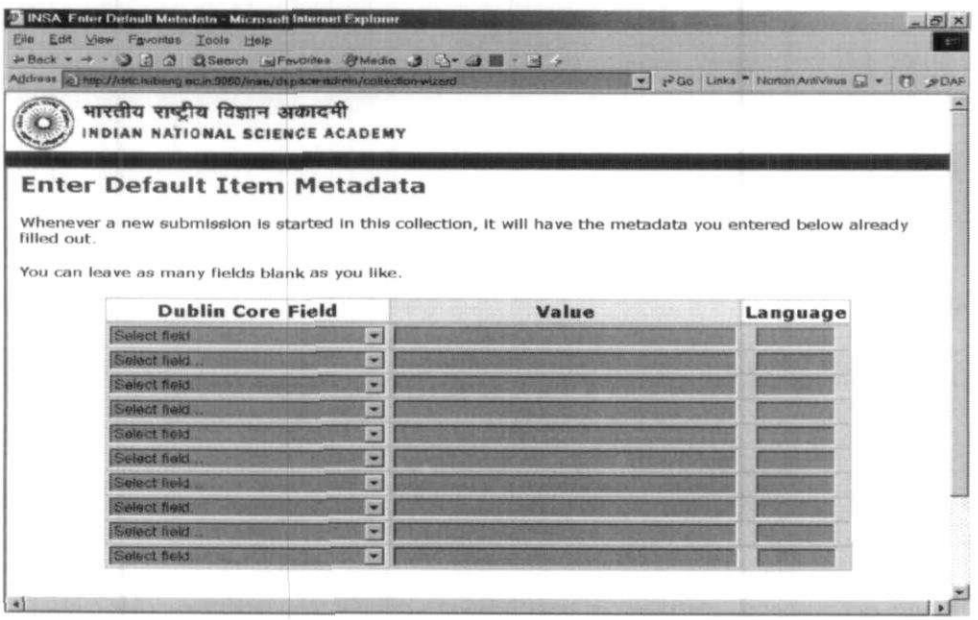


*Figure 15: Setting default values of metadata*

### Modifying Collection Description

The DSpace administrator or the collection administrator can always modify a collection description. He can remove some of the workflows, can add or delete members in the list of submitter, reviewers, metadata editors and approvers. Any modification to an existing collection should be updated by clicking 'update' button.

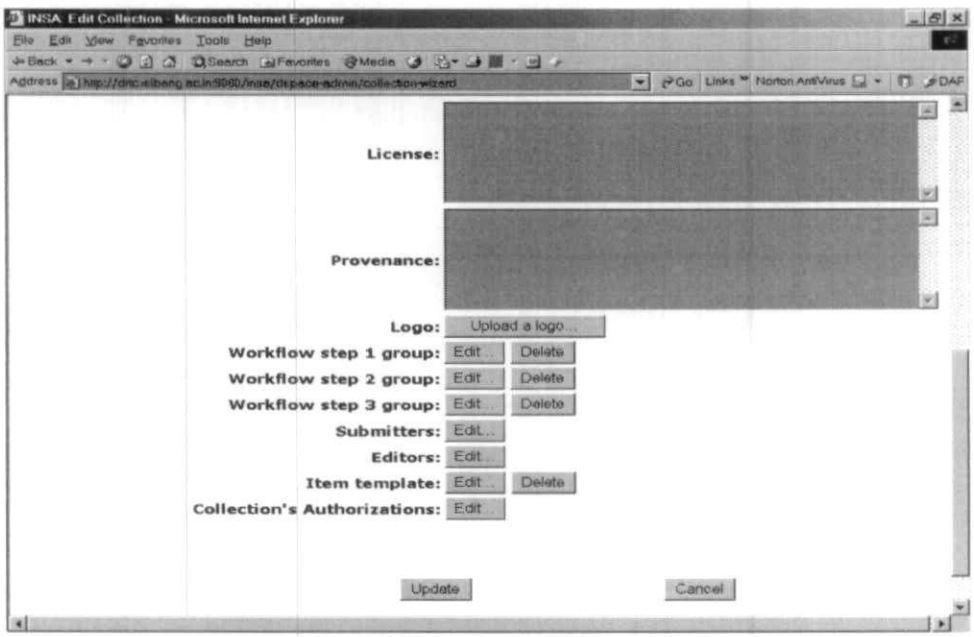


*Figure 16: Updating collection description*

Creating communities and collections is an essential step in launching a digital repository. However, optionally the DSpace administrator can modify Dublin Core registry, bitstream format registry and can perform various other tasks as and when required.

**Exercises**

1) Collect e-mail addresses of other participants and create them as members of your digital repository
2) Create the following communities:
    a. Physics
    b. Chemistry
    c. Medicine
3) Create the following sub-communities under 'Medicine' community
    a. Pediatrics
    b. Orthopedics
    c. Gynecology
4) Create the following collections under the community 'Physics'
    a. Published articles
    b. Pre-prints
    c. Theses and Dissertations
5) Modify the community names in exercise 2, so that they appear in the following order
    a. Medicine
    b. Chemistry
    c. Physics
6) Modify the names of e-groups to mnemonic names, like the following
    a. Physics_theses_submitters
    b. Physics_theses_approvers
    c. Physics_theses_reviewers
7) Add some e-persons to the above groups mentioned in exercise 6
8) Delete some e-persons to the above groups mentioned in exercise 6
9) Add default metadata elements for type and language of a collection
10) Remove workflow 1 and workflow 2.
11) Upload a logo to collection
12) Upload a logo to a community
13) Create a collection Administrator
14) Check the collection authorization and see the permissions are given according to the way you wish to have authorizations.
15) Enter a meaningful side bar text to a collection and check whether it is displayed properly in the collection page.

# A Tutorial on Dublin Core

**Devika P. Madalli**
*Documentation Research and Training Centre*
*Indian Statistical Institute*
*Bangalore*
*devika@drtc.isibang.ac.in*

## Introduction

The simplest definition of metadata is "structured data about data". Metadata is descriptive information about an object or resource whether it is physical or electronic. While metadata itself is relatively new, the underlying concepts behind metadata have been in use for as long as collections of information have been organized. Library card catalogs represent a well-established type of metadata that has served as collection management and resource discovery tool for decades. Metadata can be generated either "by hand" or derived automatically using software.

## Dublin Core

The Dublin Core Metadata Initiative (DCMI) is an organization dedicated to fostering the widespread adoption of interoperable metadata standards and promoting the development of specialized metadata vocabularies for describing resources to enable more intelligent resource discovery systems.

The Dublin Core Metadata Element Set (DCMES) was the first metadata standard deliverable out of the DCMI [IETF RFC 2413 http://www.ietf.org/rfc/rfc2413.txt]. DCMES provides a semantic vocabulary for describing the "core" information properties, such as "Description" and "Creator" and "Date".

Dublin Core metadata is used to supplement existing methods for searching and indexing Web-based metadata, regardless of whether the corresponding resource is an electronic document or a "real" physical object.

Web pages are one of the most common types of resources to utilize the Dublin Core's descriptions, usually within HTML's meta tags. However increasingly there are many digital archives of physical objects that are starting to make use of the Dublin Core.

**Dublin Core and HTML**

The Dublin Core [DC1] is a small set of metadata elements for describing information resources. This paper explains how these elements are expressed using the META and LINK tags of HTML [HTML4.0].

HTML is not case-sensitive so it does not matter if you enter the DC elements in either CAPS or small letters. But, it is advisable to be consistent in this matter, in case the metadata needs to be transported to an XML file, since XML is case-sensitive. For example, in XML *'author'* is different from *'Author'*. Though HTML is currently in wide use, once standardized, eXtensible Markup Language [XML] in conjunction with the Resource Description Framework [RDF] promises to become a significantly more expressive means of encoding metadata.

**Dublin core elements**

The 15 core elements of Dublin core are -- Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights. The latest additions are the Audience and rightsHolder elements. These 17 elements are described below.The definitions are given as provided as W3C standard (http://www.w3.org/wiki/DublinCore)

**Title**

| **Title** | |
|---|---|
| Label: | Title |
| Definition: | A name given to the resource. |
| Comment: | Typically, a Title will be a name by which the resource is formally known. |

*Examples:*
<DC:Title> The Picture of Dorian Gray</DC:Title>
<DC:Title> Prolegomena to Library Classification</DC:Title>

**Creator**

| Creator | |
|---|---|
| Label: | Creator |
| Definition: | An entity primarily responsible for making the content of the resource. |
| Comment: | Examples of a Creator include a person, an organisation, or a service. Typically, the name of a Creator should be used to indicate the entity. |

*Examples:*
<DC:Creator> Jeffrey Archer</DC:Creator>
<DC:Creator> Oscar Wilde</DC:Creator>
<DC:Creator> Library of Congress</DC:Creator>

**Subject**

| Subject | |
|---|---|
| Label: | Subject and Keywords |
| Definition: | The topic of the content of the resource. |
| Comment: | Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme. |

*Examples:*
<DC:Subject> Library Classification</DC:Subject>
<DC:Subject scheme="MESH"> Paediatrics</DC:Subject>

**Description**

| Description | |
|---|---|
| Label: | Description |
| Definition: | An account of the content of the resource. |

Comment: Description may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.

*Examples:*
<DC:Description.abstract> The author presents a tutorial introduction to Perl programming examples with extensive examples on regular expressions.
</DC:Description.abstract>

<DC:Description.toc> Introduction; Vertebrates; Invertebrates; Molluscs</DC:Description.toc>

**Publisher**

**Publisher**

Label: Publisher
Definition: An entity responsible for making the resource available
Comment: Examples of a Publisher include a person, an organisation, or a service. Typically, the name of a Publisher should be used to indicate the entity.

*Examples:*
<DC:Publisher> MIT Press</DC:Publisher>

<DC:Publisher> Dell Computers</DC:Publisher>

**Contributor**

**contributor**

| | |
|---|---|
| Label: | Contributor |
| Definition: | An entity responsible for making contributions to the content of the resource. |
| Comment: | Examples of a Contributor include a person, an organisation, or a service. Typically, the name of a Contributor should be used to indicate the entity. |

*Examples:*
<DC:Contributor.Artist> Laxman, R.K.</DC:Contributor.Artist>
<DC:Contributor.Editor> M.J. Akbar</DC:Contributor.Editor>

**Date**

**Date**

| | |
|---|---|
| Label: | Date |
| Definition: | A date associated with an event in the life cycle of the resource. |
| Comment: | Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [W3CDTF] and follows the YYYY-MM-DD format. |

*Examples*
<DC:Date.created> 1990-05-14</DC:Date.created >
<DC:Date.issued scheme = "W3CDTF"> 1998-05</DC:Date.issued>
<DC:Date.issued scheme = "W3CDTF"> 1998</DC:Date.issued>

**Type**

**Type**

| | |
|---|---|
| Label: | Resource Type |
| Definition: | The nature or genre of the content of the resource. |
| Comment: | Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the DCMI Type Vocabulary [DCMITYPE]). To describe the physical or digital manifestation of the resource, use the Format element. |

*Examples:*
<DC:Type scheme = "DCMI Type"> dataset</DC:Type>
<DC:Type scheme = "DCMI Type"> software</DC:Type>

**Format**

**format**

| | |
|---|---|
| Label: | Format |
| Definition: | The physical or digital manifestation of the resource. |
| Comment: | Typically, Format may include the media-type or dimensions of the resource. Format may be used to determine the software, hardware or other equipment needed to display or operate the resource. Examples of dimensions include size and duration. Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types [MIME] defining computer media formats). |

*Examples:*
<DC:Format" scheme = "IMT">text/xml</DC:Format>
<DC:Format.medium>CD-ROM</DC:Format.medium>
<DC:Format.extent>14 minutes</DC:Format.extent>
<DC:Format.extent>856kb</DC:Format.extent>

**Identifier**

**identifier**

| | |
|---|---|
| Label: | Resource Identifier |
| Definition: | An unambiguous reference to the resource within a given context. |
| Comment: | Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Example formal identification systems include the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN). |

*Examples:*
<DC:Identifier scheme= "URI"> http://www.google.com/"</ DC:Identifier >

<DC:Identifier scheme = "ISBN"> 1-56592-149-6</ DC:Identifier >

**Source**

| Source | |
|---|---|
| Label: | Source |
| Definition: | A reference to a resource from which the present resource is derived. |
| Comment: | The present resource may be derived from the Source resource in whole or in part. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system. |

*Examples:*
<DC:Source> Images from page 54 of 1978 edition of Bernard Shaw's Saint Joan</ DC:Source >
<DC:Source> http://abc.org/xyz/</DC:Source>

**Language**

| Language | |
|---|---|
| Label: | Language |
| Definition: | A language of the intellectual content of the resource. |
| Comment: | Recommended best practice is to use RFC 3066 [RFC3066], which, in conjunction with ISO 639 [ISO639], defines two- and three-letter primary language tags with optional subtags. Examples include "en" or "eng" for English, "akk" for Akkadian, and "en-GB" for English used in the United Kingdom. |

*Examples:*
<DC:Language> Primarily English, with some abstracts also in French</DC:Language>

**Relation**

| Relation | |
|---|---|
| Label: | Relation |
| Definition: | A reference to a related resource. |
| Comment: | Recommended best practice is to reference the resource by means of a string or number conforming to a formal |

identification system.

*Examples:*

<DC:Relation.IsBasedOn> Shakespeare's Romeo and Juliet
</DC:Relation.IsBasedOn>

<DC.Relation.IsPartOf> http://abc.org/xyz/proceedings/1998/
</DC.Relation.IsPartOf>

<DC.Relation.References> urn:isbn:1-56592-149-6
</DC.Relation.References>

**Coverage**

**Coverage**

| | |
|---|---|
| Label: | Coverage |
| Definition: | The extent or scope of the content of the resource. |
| Comment: | Coverage will typically include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [TGN]) and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of coordinates or date ranges. |

*Examples*

<DC.Coverage.spatial> Columbus, Ohio, USA;
Lat: 39 57 N Long: 082 59 W </DC.Coverage.spatial>

<DC.Coverage.temporal>
US civil war era; 1861-1865 </DC.Coverage.temporal>

**Rights**

| Rights | |
|---|---|
| Label: | Rights Management |
| Definition: | Information about rights held in and over the resource. |
| Comment: | Typically, a Rights element will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the Rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource. |

*Examples:*
<DC.Rights.accessRights> only subscribers can view the resource </DC.Rights.accessRights>

<DC.Rights.license> http://www.xyz.org/license </DC.Rights.license>

**Audience**

| Audience | |
|---|---|
| Label: | Audience |
| Definition: | A class of entity for whom the resource is intended or useful. |
| Comment: | A class of entity may be determined by the creator or the publisher or by a third party. |

*Examples:*
<DC.Audience.educationLevel> elementary school students </DC.Audience.educationLevel>

<DC.Audience.educationLevel> ESL teachers
</DC.Audience.educationLevel>

**RightsHolder**

| rightsHolder | |
|---|---|
| Label: | Rights Holder |
| Definition: | A person or organization owning or managing rights over the resource. |
| Comment: | Recommended best practice is to use the URI or name of the Rights Holder to indicate the entity. |

*Examples:*
<DC.RightsHolder> Stuart Weibel</DC.RightsHolder>
<DC.RightsHolder> University of Bath</DC.RightsHolder>

**Dublin Core in DSpace**

DSpace is currently using a qualified version of the Dublin Core schema based on the Dublin Core Libraries Working Group Application Profile (LAP), known as DC-Lib. Some qualifiers were also added to suit DSpace needs. The following section illustrates the usage of qualified DC in DSpace as compared to the original Dublin Core set as well as DC-Lib:

The adaptations that can be observed in DSpace are as follows:

- The element Contributor is used instead of Creator. It has qualifiers *'advisor'*, *'author'*, *'editor'*, *'illustrator'* and *'other'*. Whereas, neither DC nor DC-Lib have qualifiers for the metadata Contributor.
- The Date element has an additional qualifier *'date.accessioned'* which is used to mention the date on which the resource was included in DSpace. Whereas, date qualifiers like *'valid'* and *'modified'* found in DC set and *'dateCaptured'* of DC-Lib set are not used in DSpace.
- An additional qualifier *'govdoc'* is added to the Identifier element in DSpace to suit the specific needs of an archive or a repository.
- Description element in DSpace has additional qualifiers like *'provenance'*, *'sponsorship'*, *'statementofresponsibility'*. Of these the first two are Administrative Metadata.
- The Relation element in DSpace does not have the qualifiers *'hasFormat'*, *'references'* and *'conformsTo'*. And it has an additional qualifiers not found in either of the other two sets: *'ispartofseries'* and *'isbasedon'*
- The elements Creator and Source are not to be used. These are meant only for harvested metadata. The reason for not using Creator is to relieve the burden from the

user to decide which entities are "primarily" responsible for the content of the resource. Instead, Contributor element is given refinements like advisor, author, illustrator, editor, etc.

**References and further reading**


Kunze (J.). Encoding Dublin Core Metadata in HTML. Request for Comments: 2731, December 1999. (Dublin Core examples are extensively barrowed from the RFC).

DCMI Metadata Terms http://dublincore.org/documents/dcmi-terms/

Library Application Profile
http://dublincore.org/documents/library-application-profile/

DSpace Metadata *http://dspace.org/technology/metadata.html*

Cataloging Electronic Resources—Olson manual. http://www.library.cornell.edu /tsmanual/CIRM/Intro.html

Review of metadata: a survey of current resource description formats. http://www.ukoln.ac.uk/metadata/desire/overview/rev_ti.htm

Milstead (Jessica) and Feldman (Susan). Metadata cataloguing by any other name by Online, January, 1999.

# Lucene Search Engine

**Devika P. Madalli**
Documentation Research & Training Centre
Indian Statistical Institute
Bangalore.
*devika@drtc.isibang.ac.in*

## Introduction

Dspace uses Lucene Search Engine, which is a part of Apache Jakarta Project. Jakarta Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search. It is an open source project available for free download from Apache Jakarta.

## Lucene in Dspace

Lucene search engine provides both browse and search facilities. These facilities are described in detail below.

## Browsing in Dspace

Browse allows you to go through a list of items in some specified order. Dspace allows you to browse through

- Community/Collection,
- by Title,
- by Author and
- by Date

***Browse by Community/Collection*** takes you through the communities in alphabetical order and allows you to see the collections within each community.

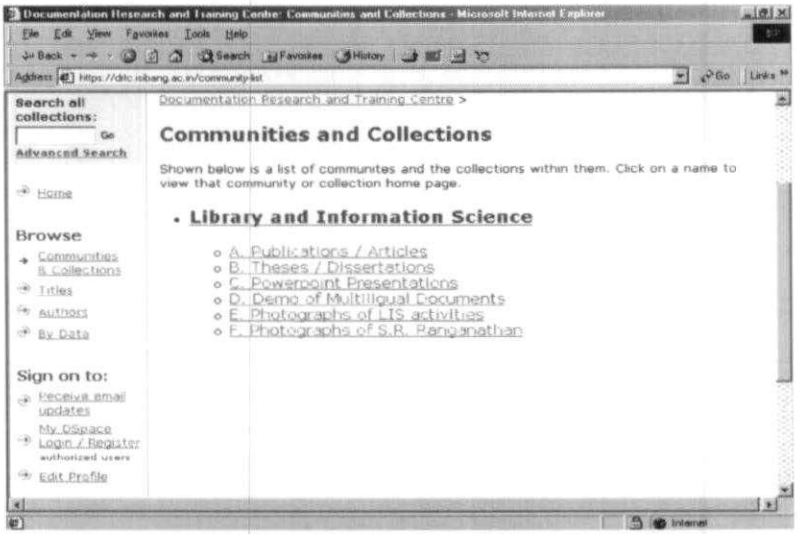


*Fig.1 Browse by Communities/Collections*

***Browse by Title*** allows you to move through an alphabetical list of all titles of items in DSpace.

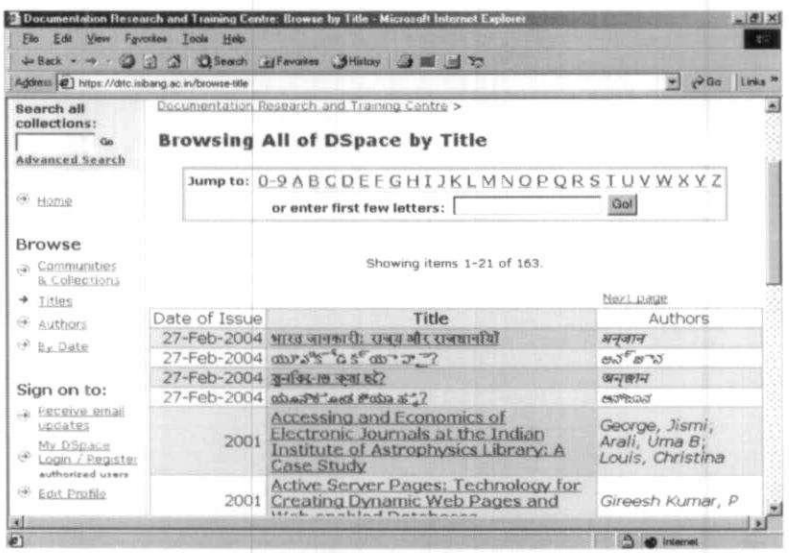


*Fig. 2 Browse by Title*

***Browse by Author*** allows you to move through an alphabetical list of all authors of items in DSpace.

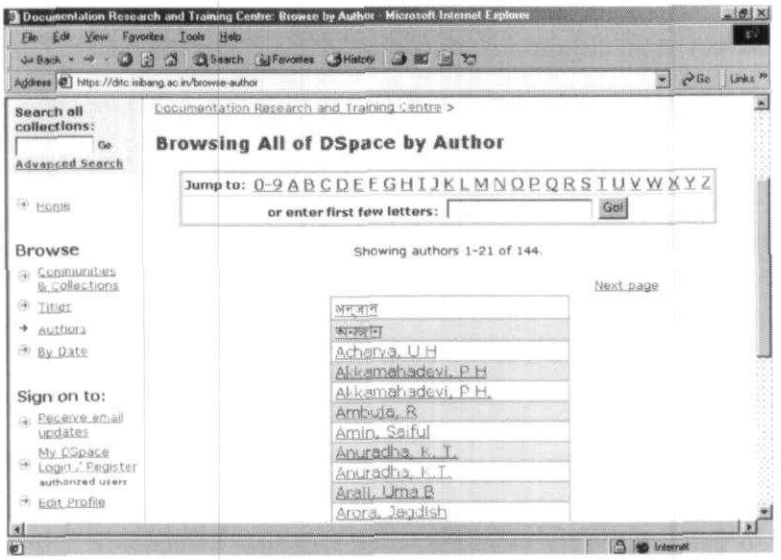


*Fig. 3. Browse by Author*

***Browse by Date*** allows you to move through a list of all items in DSpace in reverse chronological order.

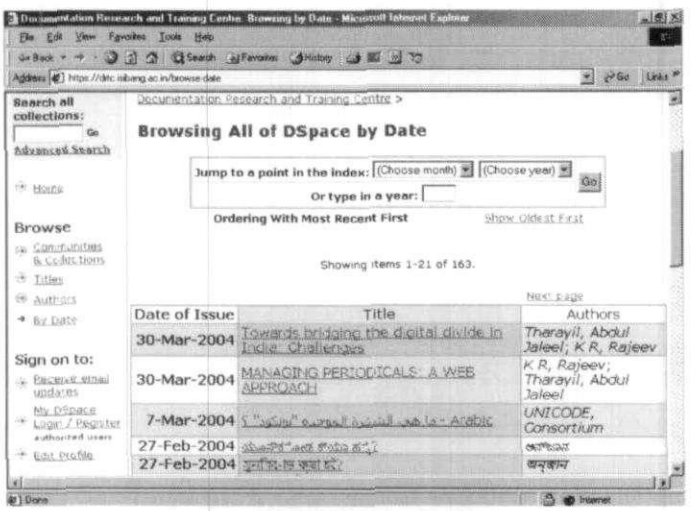


*Fig. 4. Browse by Date*

## Searching in Dspace

To search all of DSpace, use the yellow search box at the top of the navigation bar on the left. The word(s) you enter in the search box will be searched against the title, author, subject abstract, series, sponsor and identifier fields of each item's record.

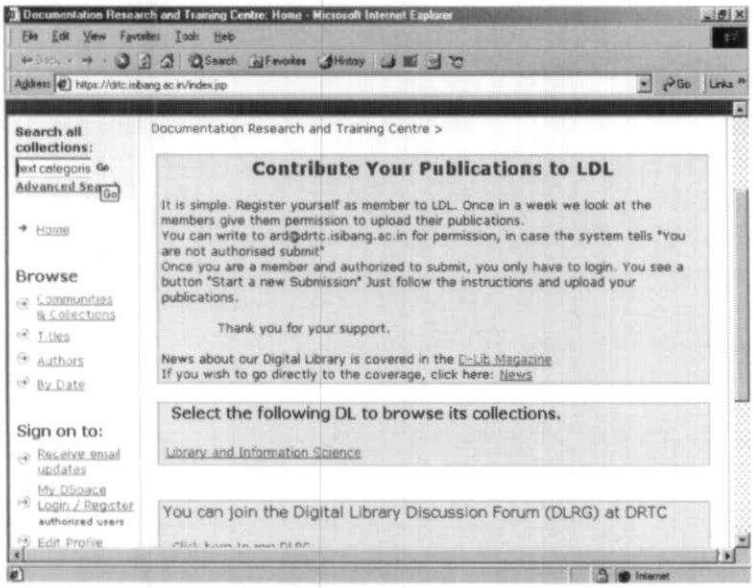


*Figure 5: searching in specific collection/community*

To limit your search to a specific community or collection, navigate to that community or collection and use the search bar on that page.

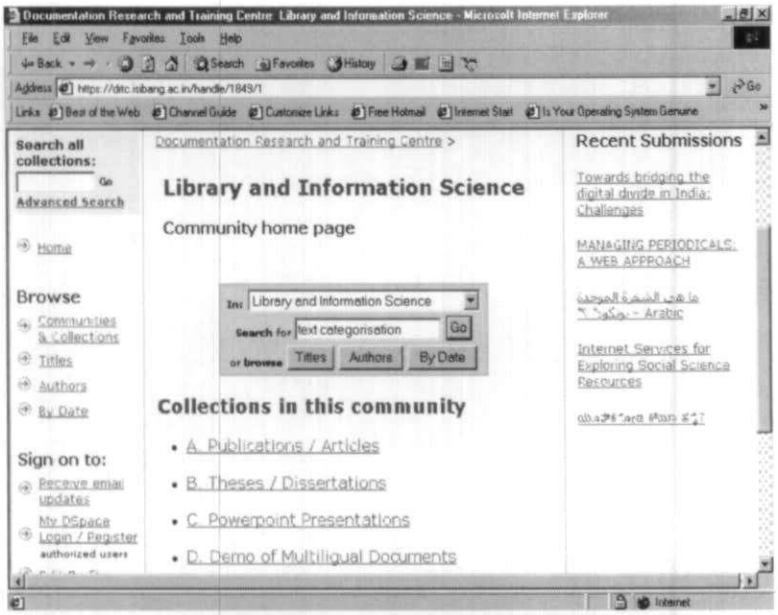


*Figure 6: Community Home Page*

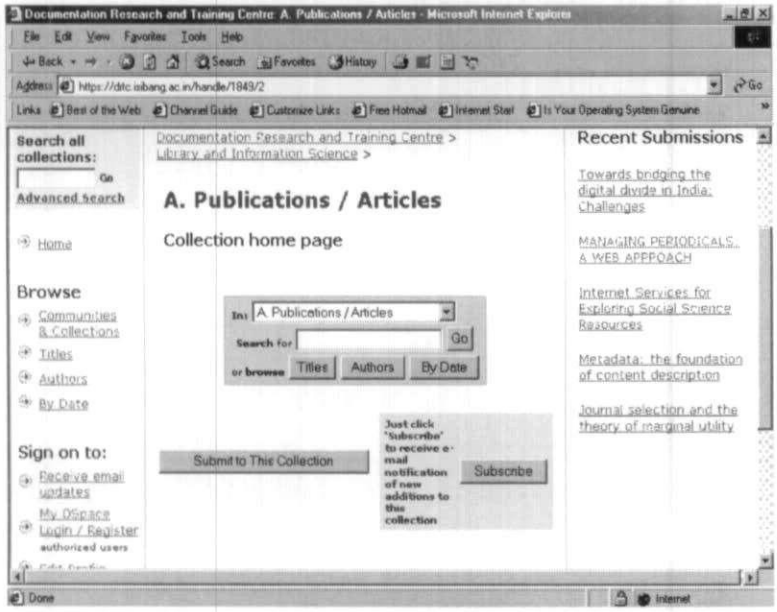


*Figure 7: Collection Home Page*

The search engine ignores certain words that occur frequently in English, but do not add value to the search. These are:
"a", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in", "into", "is","it","no", "not", "of", "on", "or", "such", "the", "to", "was"

**Kinds of search Queries in Dspace**

The syntax of the queries is given below.

**Exact Term/Phrase Search**

The search term can be a word or a phrase. One can use a search word, e.g. "information" or a phrase "information retrieval". For phrase search, the phrase should be enclosed with double quotes.

Put a plus (+) sign before a word if it MUST appear in the search result. For instance, in the following search the word "science" is optional, but the word "library" must be in the result.

e.g. +library science

Put a minus (-) sign before a word if it should not appear in the search results. Alternatively, you can use NOT. This can limit your search to eliminate unwanted hits. For instance, in the searches

e.g. planning – management
planning NOT management

you will get items containing the word "planning", except those that also contain the word "management".

**Field Search**

One can search for a term in a particular field. For example,

author:jaba
title:web
keyword:ocr
handletext:13 (brings the document having the handle number 13 in the result)
abstract:digital
mimetype:msword
sponsor:ala

**Wild cards & Stemming**
The symbol '?' is used for a single character, as in 'te?t' that matches words like 'test', 'text' etc. The symbol '*' is used for multiple characters matching, as in "inf*" matches with information, informetrics, etc.

The search engine automatically expands words with common endings to include plurals, past tenses, etc.

**Fuzzy Search**
One of the popular fuzzy search algorithms is Levenshtein distance algorithm named after the Russian scientist Vladimir Levenshtein, who devised the algorithm in 1965. It is also called 'Edit Distance algorithm'.

Levenshtein Distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t. For example,

- If s is "test" and t is "test", then LD(s,t) = 0, because no transformations are needed. The strings are already identical.
- If s is "test" and t is "tent", then LD(s,t) = 1, because one substitution (change "s" to "n") is sufficient to transform s into t.

The Levenshtein distance algorithm has been used in:
- Spell checking
- Speech recognition
- DNA analysis
- Plagiarism detection

In Dspace implementation, one can use in the following way:

Example: author:sanker~
can match shankar

You can notice, the search word has 'sa' not 'sha' and also 'ker' not 'kar'.

**Proximity Search**
Proximity search is used in a query to retrieve documents that have two words or phrases in proximity i.e. that they appear near to each other.

*"information system"~3*

Retrieves records where the words *'information'* and *'system'* are within the three words distance. Thus the above search retrieves the following titles.

*Thesaurus in an Automated Information Retrieval System*
*International Nuclear Information System: An Overview*

**Range search**

If the search query is:

*author:[prasad to rao]*

Then the system retrieves documents authored by names that fall between *'prasad'* and *'rao'*.

Whereas, the query *'author:{prasad to rao}'* **excludes** *Prasad* and *Rao*

**Boosting a Term**

Lucene provides the relevance level of matching documents based on the terms found. To boost a term use the caret, "^", symbol with a boost factor (a number) at the end of the term you are searching. The higher the boost factor, the more relevant the term will be the search result. Boosting allows you to control the relevance of a document by boosting its term. For example, if you are searching for 'Internet web'

and you want the term *"internet"* to be more relevant, boost it using the ^ symbol along with the boost factor next to the term. You should type:

*internet^5 web*

By default, the boost factor is 1. Although the boost factor must be positive, it can be less than 1 (e.g. 0.2)

**Boolean Search**

Boolean 'AND', 'OR', 'NOT' are used for Boolean combinations. Boolean operators should be in caps.

- 'OR' is the default conjunction operator. One can use '||' instead of 'OR'.
- Either 'AND' or '&&' can be used for Boolean 'AND'.
- Either 'NOT' or '!' can be used for Boolean 'NOT'.

Examples,

- ***"library science" AND "information science"*** matches documents where both terms exist anywhere in the text of a single document

- ***"library science" OR "information science"*** links two terms and finds a matching document if either of the terms exist in a document

- ***"library science" NOT "information science"*** excludes documents that contain the term after NOT, in this case it retrieves documents that do not contain the term "information science"

**Group Search**

Parentheses can be used in the search query to group search terms into sets, and operators can then be applied to the whole set. For example,

*(interactive resources OR learning objects) AND (Geography)*

The above search query retrieves documents that WILL contain the term Geography and either term *interactive resources* or *learning objects* may exist in the retrieved document.

**Field Grouping**

Parentheses can be used to group multiple clauses to a single field. For example, To search for a title that contains both the word "Geography" and the phrase " interactive resources" use the query:

*title: (+"interactive resources" +Geography)*

**Advanced Search**

The advanced search page allows you to specify the fields you wish to search, and to combine these searches with the Boolean "and", "or" or "not".

You can restrict your search to a community by clicking on the arrow to the right of the top box. If you want your search to encompass all of DSpace, leave that box in the default position. Then select the field to search in the left hand column and enter the word or phrase you are searching in the right hand column. You can select the Boolean operator to combine searches by clicking on the arrow to the right of the "AND" box.
You MUST use the input boxes in order. If you leave the first one blank your search will not work.

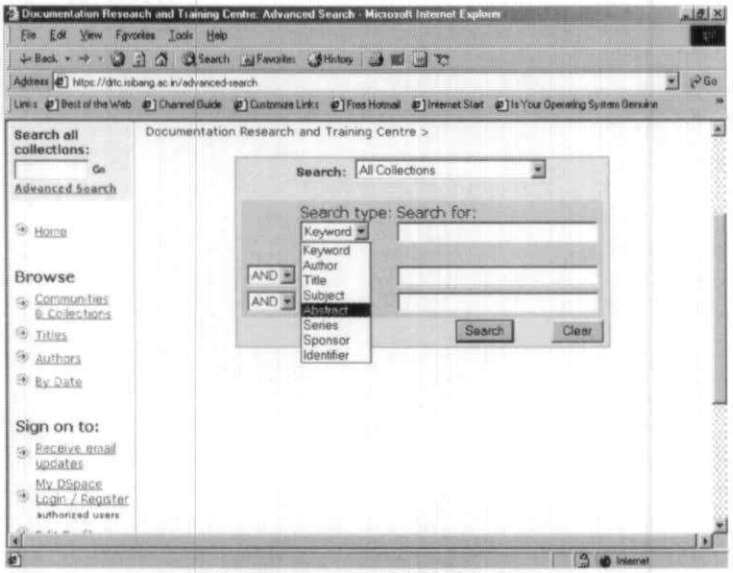


*Fig. 8 Advanced Search Page*

## References

Jakarta Lucene – Overview *http://jakarta.apache.org/lucene/docs/index.html*

Jakarta Lucene – Query Parser Syntax
*http://jakarta.apache.org/lucene/docs/queryparsersyntax.html*

Gilleland, Michael. Levenshtein Distance, in Three Flavors.
*http://www.merriampark.com/ld.htm*

# Authority Control in DSpace: For Author, Journal Title, Publisher Name

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

Traditionally, Library and Information Science professionals have been using various authority files, like, personal name, Institutional name, serials, journal title etc. This is to ensure standardization of data values, to achieve interoperability. The recent versions of DSpace has introduced some of them, especially personal (author) name, journal title and publisher name. It should be noted that this works *better with the xmlui interface*.

## Authority control settings in DSpace

For each metadata field, one can configure:

A. ChoiceAuthority plugin, which marks the field as under Choice Management.
B. Presentation style.
C. Closed or open choices.
D. Authority controlled (choice management also required).
E. Authority value required.

### Presentation Style

- **lookup** - User enters a proposed value and clicks a button to "look up" choices based on that value, a pop-up window appers and lets them navigate through choices.
- **suggest** - As the user types in a text-input field, a menu of suggested choices is automatically generated.
- **select** – Displays a drop-down menu (or multi-pick selection box) of choices using the HTML SELECT widget. This style should *only* be used for plugins that have a relatively small, and fixed set of choices. It does *not* support authority values and should not be used for authority-controlled fields.

The following table gives an idea of which input-type can go with what kind of presentation style (1)

| input-type (from input-forms.xml) | 'lookup' Presentation Style | 'suggest' Presentation Style | 'select' Presentation Style | Authority Control |
|---|---|---|---|---|
| onebox | yes | yes | yes | supported |
| twobox | yes | yes | yes | supported |
| textarea | yes | yes | yes | supported |
| name | yes | NO | NO | supported |
| date | NO | NO | NO | n/a |
| series | NO | NO | NO | not tested |
| dropdown | NO | NO | yes | NO |
| qualdrop_value | NO | NO | NO | NO |
| list | NO | NO | ??? | not tested |

*Table1: Presentation Style input*

From the above table, we can conclude:
Authority-control feature can be used only with onebox, twobox, textarea, name. This feature is not useful in case of date, dropdown, qualdrop, list input types
Perhaps, in future series may be included, as series authority is used by librarians.
**Note:** Please go to *$DSPACE_HOME/config/input-forms.xml* file, to check the input-type for a metadata field for which you intend to use authority-control.

- **Step 1:** In the latest versions i.e. 1.7.0 and above , **you can skip this step**. Check whether the database tables are already set properly to accommodate authority and confidence level, using the following commands

```
$psql dspace=> select * from
metadatavalue;
```

If you see, something like the following, where authority, confidence are in the columns, you need not do anything.

```
metadata_value_id | item_id | metadata_field_id |
text_value | text_lang | place | authority |
confidence
```

If you do not see the columns authority and confidence, you should run the following SQL command:

```
ALTER TABLE MetadataValue
    ADD COLUMN authority VARCHAR(100),
    ADD COLUMN confidence INTEGER DEFAULT -1;
```

- **Step 2: Setting up Personal Name, Publisher and Journal Title Authority**

The following lines in *SDSPACE_HOME/dspace/config/dspace.cfg* file indicate how you will configure your ChoiceAuthority plugins in the PluginManager style.

```
plugin.named.org.dspace.content.authority.ChoiceAuthority = \
  org.dspace.content.authority.SampleAuthority = Sample, \
  org.dspace.content.authority.LCNameAuthority = LCNameAuthority, \
  org.dspace.content.authority.SHERPARoMEOPublisher = SRPublisher, \
  org.dspace.content.authority.SHERPARoMEOJournalTitle = SRJournalTitle
```

This sets the tone for setting up LCNameAuthority, Sherpa-Romeo sites Publisher Name and Journal Title.

- **Step 3:** To pickup all the value-pairs elements defined in your input-forms.xml, you need have the following lines in dspace.cfg file

```
plugin.selfnamed.org.dspace.content.authority.ChoiceAuthority =
\
  org.dspace.content.authority.DCInputAuthority
```

- **Step 4:** Configuring the resources

In dspace.cfg file, uncomment the line lcname.url and sherpa.romeo.url, as shown below

```
## configure LC Names plugin
lcname.url = http://alcme.oclc.org/srw/search/lcnaf
## configure SHERPA/RoMEO authority plugin
sherpa.romeo.url = http://www.sherpa.ac.uk/romeo/api24.php
```

- **Step 5:** Every authority-controlled field must also be configured with a source of choices, presentation style

For **Personal Name Authority-control**

```
choices.plugin.dc.contributor.author =  LCNameAuthority
choices.presentation.dc.contributor.author = lookup
authority.controlled.dc.contributor.author = true
```

For **Publisher Name Authority-control**

```
choices.plugin.dc.publisher = SRPublisher
choices.presentation.dc.publisher = suggest
```

For **Journal Title Authority-control**

```
choices.plugin.dc.title.alternative = SRJournalTitle
choices.presentation.dc.title.alternative = suggest
authority.controlled.dc.title.alternative = true
```

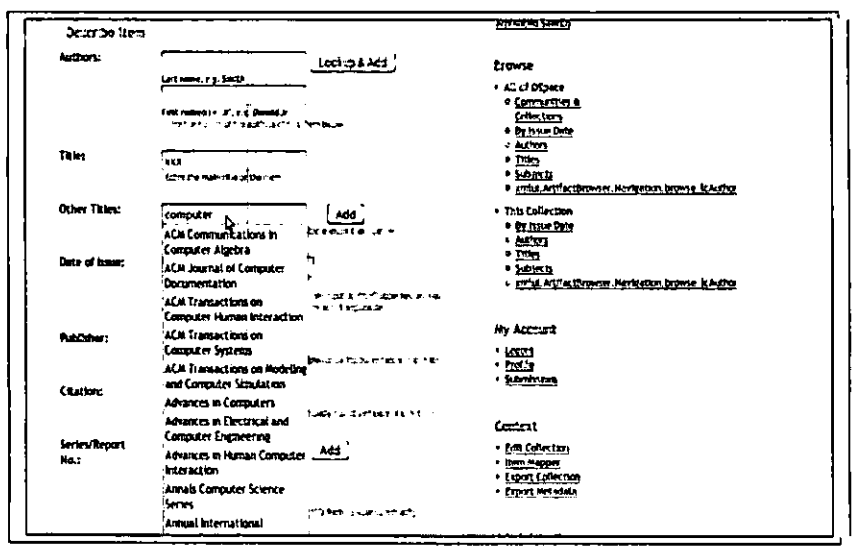


*Figure1: LC Name Authority author lookup*

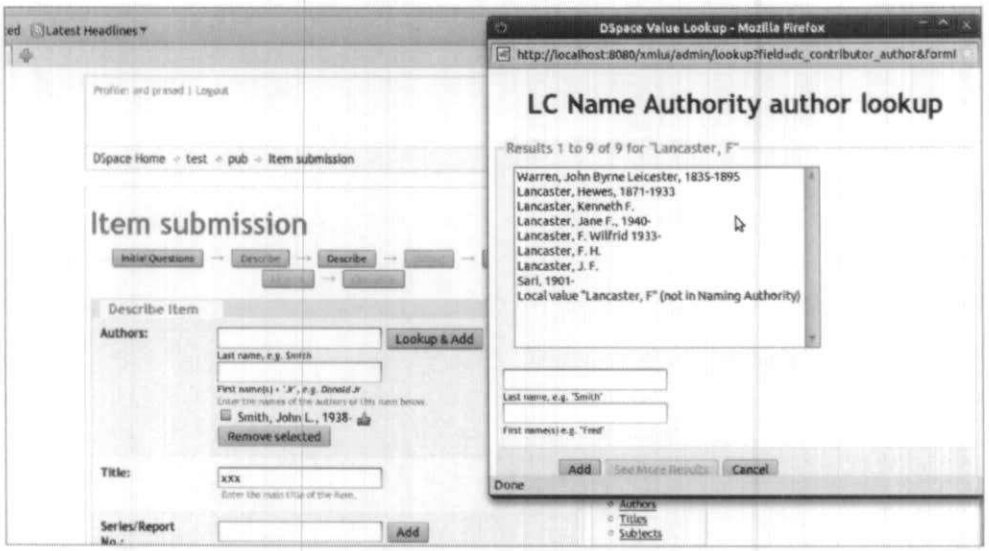


*Figure 2:Title Authority taste*

**References**

Authority Control of Metadata Values.
https://wiki.duraspace.org/display/DSPACE/Authority+Control+of+Metadata+Values

# Ontology in DSpace

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

The vocabulary control devices are age old tools in use by library and information community in the form of thesauri or classauri. In the semantic web parlance they have got transformed into either SKOS (Simple Knowledge Organization System), or OWL (Web Ontology Language). Many ontologies like AGROVOC, MeSH are available either in SKOS or OWL or both formats. Both OWL and SKOS use XML to define the structure. Though there have been attempts to implement these formats, so far the official versions of DSpace could not offer either of them. Instead, DSpace uses a basic XML file to express partially some of the relations offered by thesauri. However, the relations are restricted to BT-NT relations only.

## Vocabulary control in DSpace

- The file which allows to express BT-NT relations are found or you can create in *$DSPACE_HOME/config/controlled-vocabularies*
- The file name should have ".xml" as extension. It has only three tags: *<hasNote>*, *<isComposedBy>*, *<node>*
- The official version of DSpace comes with two sample files: *nsi.xml and srsc.xml* (English)

If you intend to *create* vocabulary file

- The file should start with the following mandatory line
  *<?xml version="1.0" encoding="UTF-8"?>*
- Followed by <node> tag, which can contain two attributes "*id*" and "*label*"
  Ex: *<node id=xxx label="Library Science">*

where xxx is a kind of arbitrary serial alpha-numeric number, which does not serve any purpose except identifying the label with a unique number/id. *Label* attribute contains the actual subject term, which is displayed as menu item in the form of hierarchy and gets indexed and can be searched. The ending tag of the node i.e. *</node>* will occur after you

have entered the narrower terms which is embedded between the tags *<isComposedBy> and </isComposedBy>*

- Each key term whether narrower or broader is represented using the tag *<node>*
- The narrower terms (NTs) are enclosed between *<isComposedBy>* and *</isComposedBy>*
- The tag *<hasNote>* does not serve any purpose, except providing information to the creator of the XML file and DSpace does not take it into its cognizance

Though the sample file has lots of indentions used quite liberally, it is not mandatory to have indentions. The main restriction is that every tag should have an ending tag. This is typically the case with all XML files.
Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<node id="root" label="Subject Terms">
   <isComposedBy>
     <node id="1" label="Library Science">
       <isComposedBy>
          <node id="11" label="Classification">
            <isComposedBy>
           <node id="111" label="Enumerative Classification">
             <hasNote></hasNote>
            </node>
            <node id="112" label="Analytico Synthetic Classification">
             <hasNote></hasNote>
           </node>
        </isComposedBy>
       <node id="12" label="Cataloguing"> </node>
      <node id="13" label="Reference Source </node>
     </isComposedBy>
   </node>
  </isComposedBy>
```

Normally, one can use software like Protege for creating OWL or SKOS file. For this kind of XML file, one has to create manually. As XML is case sensitive one has to be careful with tags. Unlike HTML, XML is not tolerant to errors and one has to take extra care in opening and closing tags.

**How to Use**

- **Step 1:** Create an xml file in *$DSPACE_HOME/config/controlled-vocabularies*. In this example, we use the existing *srsc.xml* file

- **Step 2:** Edit the dspace.cfg file and uncomment the following line *webui.controlledvocabulary.enable = true*

- **Step 3**: in the *$DSPACE_HOME/config* directory, edit the file input-forms.xml.
  For example, to see the vocabulary, under subject keywords, the entry should look like

```
<field>
   <dc-schema>dc</dc-schema>
   <dc-element>subject</dc-element>
   <dc-qualifier></dc-qualifier>
   <!-- An input-type of twobox MUST be marked as repeatable -->
   <repeatable>true</repeatable>
   <label>Subject Keywords</label>
   <input-type>twobox</input-type>
   <hint> Enter appropriate subject keywords or phrases below.</hint>
   <required></required>
   <vocabulary>srsc</vocabulary>
```

Note: Instead of *<vocabulary>srsc</vocabulary>*,you may use your own xml file having a different vocabulary. In case you do not require the parameter you need not mention in the *<field>* tag, as in the case of title, creator etc.

- **Step 4:** Stop and Start tomcat

- **Step 5:** while entering metadata in the input-form, you will see something like the following screenshot.

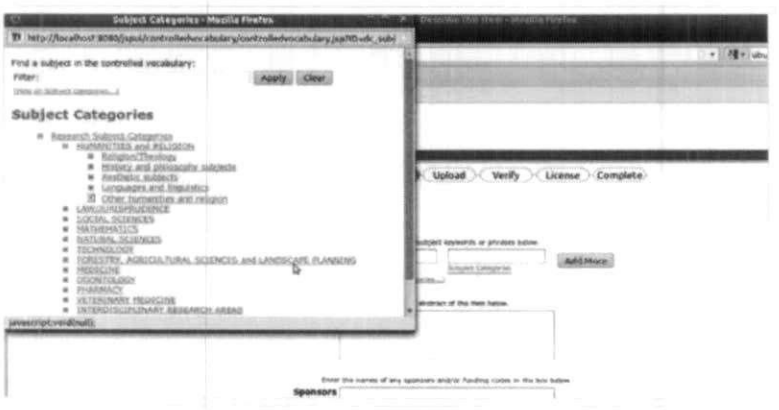


*Figure1:Subject categories*

- **Step 6:** You may click on any key term you wish, the system will add the term you have clicked and also all its broader terms separated by "::". This is normal, so that each document will have the actual term and its broader terms assigned to it and all the terms get indexed, so that when the end-user searches by a broader term, the documents having the narrower terms are retrieved.

**Conclusion**

This vocabulary plug-in works fairly fast when the vocabulary is limited. But on a very large set of vocabulary the display of the pop-up window, becomes unacceptably slow.

# Harvesting in DSpace

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

The OAI-PMH protocol is used to harvest metadata from data providers (digital repositories) by service providers (harvesters). For example, you may consider Librarian Digital Library (LDL – https://drtc.isibang.ac.in) as a data provider (source repository) and Search Digital Libraries (SDL – http://drtc.isibang.ac.in/sdl) as a service provider. SDL uses PKP harvester, which collects only Dublin core metadata from various repositories (including LDL) dealing with Library and Information Science. However, if one wishes to collect not only the metadata but also the digital objects, the data provider repository should support/enable OAI-ORE (Open Archives Intiative – Object Reuse and Exchange) and should not be on SSL (Secure Socket Layer i.e. https). How to enable OAI-ORE to allow others to capture your repository's digital objects is not covered in this paper. Here, we explain, how to harvest metadata and digital objects from others repositories. This is supported only in XMLUI not in JSPUI.

- **Step 1:** Go to XMLUI, create collection under a community. Once you fill up the information under various boxes, click 'create' button.
- **Step 2:** In the next screen, you will see a horizontal menu, having the following items *Edit Metadata Assign Roles Content Source Curate*
- Click *"Content Source"*
- **Step 3**: In the next screen you will see the following

> **Content Source:**
>
> - This is standard DSpace collection
> - This collection harvests its content from an external source

The first option is the regular option, as in the case of older DSpace versions, whereas the second option is a new feature, where you are **not** expected allow authors to upload their publications, instead you are going to populate the collection from another repository which is ORE enabled. You can choose only one option as this is a radio button.

Choose either OAI-PMH or OAI-ORE option

- **Step 4:** In the next screen you will see many options to be chosen*/filled.*

***OAI Provider:*** You have to enter the OAI base URL of the data provider repository. For LDL it is:

*http://drtc.isibang.ac.in/oai/request*

You can obtain the base URL from any of the following sites

*http://www.openarchives.org/Register/Browsesites*

*http://www.opendoar.org/countrylist.php*

***OAI Set ID:***

All Sets

Specific Sets

If you mark all sets, you do not have fill up the box below, but you mark Specific Sets, you should know the Set Id, which can be obtained using the following URL in a browser

*http://xxx.xxx.xxx?verb=ListSets*

Replace xxx.xxx.xxx by *BaseURL* of the Repository you wish to harvest. The above URL will display an XML output and look for the following tags

```
<set>
<setSpec>hdl_1_411</setSpec>
<setName>ha</setName>
</set>
```

Collect the data value of the element *<setSpec>* to be used as a Specific set.

***Metadata Format:*** In the pull-down menu, you can choose any of the items – Qualified Dublin Core, Simple Dublin Core, DSpace Intermediate Metadata

If you wish test the baseURL, Specific Sets and Metadata Schema support, click *"Test Settings"* button. If everything is OK, you should see on the top of the screen:

If you wish to harvest only the metadata as in the case of PKP harvester, mark the first option. The second option creates hyperlinks to the digital objects of the data provider/source repository, but does not populate the collection with actual objects. However, the last option populates the collection with both metadata and digital objects associated with metadata.
***Note:*** For the second and third options to work, the source repository should have OAI-ORE enabled.

**Notice**
Harvesting Setting are valid
Harvesting Options:
- Harvest metadata only.
- Harvest metadata and references to bitstreams (requires ORE support).
- Harvest metadata and bitstreams (requires ORE support).

# An Introduction to DSpace Installation in Ubuntu

**ARD Prasad**
Documentation Research and Training Centre
Indian Statistical Institute
Bangalore 56 0059

ard@drtc.isibang.ac.in

## Introduction

This article attempts to explain simple configuration of DSpace and does not cover Tomcat integration with Apache or Secure Socket layer (SSL). As DSpace is mostly configured on Linux system, the following sections are applicable to Linux, in particular Redhat Linux. The other Linux flavors may have different directory structure, so some of the system files may reside in different directories than the ones mentioned here. For example, sendmail and postgreSQL files. This is typically the case with other Unix versions like HP-UX, Solaris, or AIX. Experienced Unix administrators should not find it difficult to load DSpace in any Unix environment. Although DSpace can be installed in MS-windows operating system using Oracle or using postgreSQL in cygwin environment, it is strongly discouraged for a variety of good reasons.

**Linux:**
To load DSpace you require super user password. If you are new to Linux, take the help of your system administrator. As such DSpace installation is simple, however, it uses many Linux services like mail, postgreSQL, DNS (Domain Name Service) etc. Mail and other services are to be properly configured for a successful installation of DSpace.

## Required Linux Software and Services

1. **Mail:** DSpace heavily uses mail in
    a. Registration of new members
    b. Login
    c. Sending notification to authors, reviewers, metadata validators etc
    d. Sending message to subscribers of any new additions to a collection

2. **PostgreSQL:** This is the backend database of DSpace, which stores information about

a. Communities
b. Collections
c. Members, their e-mail addresses and passwords in encrypted form
d. E-groups (reviewers, metadata validators for each collection etc.)
e. Metadata of digital items etc.

3. **DNS:** The purpose of DNS is to get IP number for a given IP address. For example, if you try to access drtc.isibang.ac.in, it gets translated to 210.212.206.71. Of course, this is quite transparent to the end user. Most likely your system might have been using the main server as the DNS server of your organization. If you prefer a separate name for your dspace server, you have to ask your network administrator to make an entry of the new host name in the DNS configuration.

**Mail configuration:** In Linux you can use either use Exim4 or Postfix as mail server. Here, a very rudimentary Exim4 configuration is explained. All the Linux services will have daemons to run the services in the background. In case of mail, the daemon is /etc/rc.d/init.d/ exim4. To reconfigure exim4, you should run

# dpkg-reconfigure exim4-config

Fill up the information in the screens that appear.

**NOTE:** Check sending and receiving mail from your system, to make sure that mail server is working properly.

**PostgreSQL configuration**

PostgreSQL daemon is /etc/rc.d/init.d/postgresql. Its configuration files are in the directory /var/lib/pgsql/data. However, if postgres was not initiated before, the required files might not have been generated. To generate the required files, you have to run the postgres daemon. Once the required files are generated, you have to tweak the following files - /var/lib/pgsql/data/postgresql.conf and pg_hba.conf in the same directory.

Also make sure whether jdbc drivers for postgresql are installed. The jdbc driver files should be copied to dspace-source/lib directory

**Java**

Some unix installation have java compiler loaded. To install java run the following command

```
apt-get install java-6-openjdk
```

**Ant and Maven**
Apache ant is similar to 'make' of C programs. It is used to generate .jar or .war file. It is essential to have build.xml file in the directory where you run ant. The following command installs both moaven2 and ant

```
apt-get install maven2
```

**Tomcat**
Java Server Pages (JSP) is similar to ASP or PHP, a server side scripting language for Web interface. As DSpace is written in JSP, we require a servlet container. Tomcat is one of the many available servlet engines. The others are JBoss, Jetty, Resin, j2ee. JBoss is supposed a more professional and powerful software. Here, tomcat is explained, as it is one of the most popular dspace user choices.

Download Tomcat and install it in $DSPACE-HOME. Do not install tomcat of ubuntu using either apt-get or synaptic

**DSpace Installation**
Follow all the steps enumerated in DSpace documentation.

**Troubleshooting**
The following are broad tips if you run into rough weather.

**At the time of Installation**
1) If the system responds ant not found or javac not found. It could mean that ant or java are not in your path, check */dspace/.bash.rc* and modify PATH values correctly

2) Any java related problems might have been caused by not setting JAVA_HOME and JAVA_OPTS properly. Or, you do not have the right java compiler. Also make sure that the *$PATH=/opt/java/bin:$PATH*, so that even if you have another java compiler, the one loaded in /opt directory gets priority.

And re-login to dspace, to take the environment variables like PATH are set properly.
You can check any environment variable with the following command
echo $PATH or
echo $JAVA_HOME
echo $JAVA_OPTS

3) while running 'ant fresh_install' if you get SQL related errors, it could mean either
a) postgreSQL configuration was not done properly or

b) the jdbc drivers are not copied lib directory of your DSpace source directory

**NOTE:** This is by far the most common problem, while installing DSpace

**While running DSpace**
Once installation is complete, open DSpace in the browser.
If you do not see DSpace screen, check whether tomcat is running. For example:
http://drtc.isibang.ac.in:8080 should show tomcat
http://drtc.isibang.ac.in:8080/dspace should show DSpace screen

In all likelihood 8080 port is already being used by some other service or another tomcat running on the same port. In which case, you have to choose another port for your tomcat. This can be achieved by replacing 8080 and the related ports in tomcat/conf/server.xml file.

If you see the DSpace running, it is good idea to check whether it is sending mail to newly registered members. If you get internal system error message, it could mean that mail server is not configured properly. Also check in the dspace.cfg file, the mail.server variable is properly set.

If DSpace is sending mail, but the link given the mail is not accessible on the browser means, in the dspace.cfg file, the host name is not specified properly. It may be having https, though you are not using SSL.

If you get "Internal system Error" while creating communities, it could mean postgreSQL was not configured properly.

**LOG Files:**
"Internal system error" is too generic an error, it might have been caused by any configuration lapse. To locate the exact problem, go through the tomcat/logs/catalina.out file and also /dspace/log/dspace.log file

**Check from out side:**
If everything is working fine, try to access DSpace from another system on your LAN and also from another system on internet. If DSpace is working on your LAN but not from outside, you are behind a firewall. You may to ask your administrator to open 8080 port in the firewall.

**Summary of Steps**

If you see '#' prompt in the following commands, you should run as root/super user.
Step 1: create dspace user account, run
# adduser dspace –home=/dspace

Step 2: create dspace as postgresql database user
# su – postgres
$ createuser -U postgres -d -A -P dspace

provide 'dspace' as password. If you chose any other password, you have to enter the same password in /dspace/dspace-1.7.2-release/dspace/config/dspace.cfg before running 'maven'

Step 3: copy the following downloaded files in /dspace directory

apache-tomcat-7.0.14.tar.gz
dspace-1.7.2-release.tar.gz

Step 4: untar and unzip the above files
# su – dspace
$ tar zxvf apache-tomcat-7.0.14.tar.gz
$ tar zxvf dspace-1.7.2-release.tar.gz

Step 5: create dspace database
$ createdb -U dspace -E UNICODE dspace

Step 6: change the current directory and run maven using the following commands

$ cd dspace-1.7.2-release/dspace
$ mvn package

This will take lot of time, as this step downloads lots of required files from the Internet. Once you build successfully, run the following commands

$ cd /dspace/dspace-1.7.2-release/dspace/target/dspace-1.7.2-build.dir
$ ant fresh_install

Step 7: copy the generated file to tomcat

cp -r /dspace/webapps/* /dspace/apache-tomcat-7.0.14/webapps

Step 8: run tomcat

/dspace/apache-tomcat-7.0.14/bin/startup.sh

Step 9: run
firefox http://localhost:8080/jspui

**Further help:**
You can join the DSpace Mailing Lists mentioned at DSpace home page: http://www.dspace.org. You can also join DRTC DL mailing list at http://drtc/isibang.ac.in/dlrg

**Sites for downloading required files:**

| | |
|---|---|
| **ant:** | http://ant.apache.org |
| **tomcat:** | http://jakarta.apache.org |
| **java:** | http://java.sun.com/j2see |
| **dspace:** | http://www.sourceforge.net/projects/dspace |
| **dspace Homepage:** | http://www.dspace.org |

# Interoperability and the OAI-PMH

**ARD Prasad**
Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

With the growing number of digital repositories in the Web, it has became difficult for the users to visit individual places in search of information. Many organizational repositories have not been indexed by the search engines. Such mechanism is therefore required by which the repositories can share the resources and work in coordination, to provide a broader purview to the users. The mechanism which provides the ability to the information systems to work in coordination has been termed as *Interoperability.* Open Archives Initiative is one of the landmark efforts to ensure the availability of the metadata of digital resources of many repositories at the users' end.

## About Interoperability and Open Archives Initiative

In most generic way, interoperability is the ability of systems, organizations and individuals to operate together to achieve a common goal. In digital libraries *search interoperability* is the main concern. Priscilla Caplan (2003) defined *Search Interoperability* as 'the ability to perform a search over diverse set of metadata records and obtain meaningful results'. To bring such interoperability among digital libraries, remarkable effort has been made by Open Archives Initiative (OAI). Open archives initiative develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content. The Initiative is supported by the Digital Library Foundation, the Coalition for Networked Information and National Science Foundation. The Open Archives Initiative has its roots in an effort to enhance access to e-print repositories (archives) as a means of increasing the availability of scholarly communication.

### *Prerequisites to develop metadata harvesting protocol*

To facilitate metadata harvesting there needs to be agreement on:

- Transport protocol - HTTP or FTP or other such protocol
- Metadata format - Dublin Core or MARC or other such format

- Metadata Quality Assurance - mandatory element set, naming and subject conventions, etc.
- Intellectual Property and Usage Rights - *who* can do *what* with *what*?

**About OAI-PMH**

The Open Archives Initiative Protocol for Metadata Harvesting provides an application-independent interoperability framework based on *metadata harvesting* (OAI- PMH,Version 2.0 )that can be used by a variety of communities who are engaged in publishing content on the Web

It provides a set of rules that defines the communication between systems (like FTP or HTTP in Internet). That's why even though the protocol actually uses HTTP as a transport mechanism between digital libraries, it is popularly known as the "*HTTP of Digital Libraries*" (Rhyno, 2004)

The protocol was developed with an objective to ensure interoperability between eprint repositories only, later in version 1.0/1.1 all document like digital objects also brought into its purview and finally the latest version 2.0 supports all kinds of digital resources. The protocol is based on HTTP and XML. But the greatest limitation is that it doesn't support the previous (older) versions.

**Why use OAI-PMH**

- Implementation is quite simple as compared to distributed search protocol (Z39.50).
- Search process is faster in centralized search mechanism because the metadata collected from different repositories are kept at local server and retrieved from there at the time of search. Whereas in distributed search mechanism (Z39.50) metadata is retrieved from different repositories at the time end-user submits a search query.
- The service providers can modify the display of collected metadata as per their metadata policy.

**How OAI-PMH Works (Structural and Functional Model)**

The OAI-PMH uses HTTP as Internet protocol and supports both GET and POST HTTP verbs (request methods). GET retrieves information in any format whereas POST requests to pass the information with HTML forms. XML for encoding and exchanging information and qualified and simple Dublin Core as metadata schema, still it supports other metadata schemas like MARC21, EAD, ONIX, METS, etc.

***Key Players in OAI-PMH* (OAI- PMH Version 2.0)**

- *Service Provider*: A service provider issues OAI-PMH requests to data providers and uses the metadata as a basis for building value-added services. These repositories send requests using 6 OAI-PMH verbs
- *Data Provider*: A data provider maintains one or more repositories (web servers) that support the OAI-PMH as a means of exposing metadata. These are the repositories which process the request and respond to service providers with appropriate OAI-PMH responses.

The protocol divides the whole universe into two i.e. Service providers or metadata harvesters are the repositories (web servers) that issues requests to other repositories to get the metadata. There is a standard set of 6 OAI-PMH verbs (request syntax) used to send request to data providers. The request is transmitted according to the rules of HTTP over the Web. Data providers receive these requests and reply with appropriate OAI-PMH responses in valid XML format specified by the OAI-PMH protocol. Thus while receiving the response, the service provider ca understand who is the metadata provider (*Identify* verb), what is the metadata format (*ListMetadataFormat* verb), and what are different sets or divisions of that metadata format (*ListSets* verb). The service provider can get specific metadata by using the verbs like *GetRecords*, *ListIdentifiers*, *ListRecords*. Remember that a repository can act both as service provider as well as data provider, or only a data provider or service provider.

***Some facts about OAI-PMH***

- OAI-PMH is not a search engine, or a search tool, or a database.
- It only provides set of rules to move the metadata not the content of the digital resource from one repository to another. The content remains in the source repository only.
- A repository can act both as service provider (or harvester) and data provider or only service provider or data provider
- The protocol is not restricted only to support simple (unqualified Dublin Core), but can support any metadata schema which can be provided in XML format.

***Features of OAI-PMH* (OAI – PMH Version 2.0)**

- **Flow Control (resumptionToken):** OAI-PMH requests may return list of records, headers (unique identifiers) and sets. In some cases data providers send incomplete lists with resumptionTokens so that the service provider can issue further requests to complete the list. Following optional attributes can be added as resumptionToken elements – expirationDate, completeListSize, cursor.

- **Error and exception conditions:** In case of error or exception conditions, repositories indicates OAI-PMH errors using following Error Codes – badArgument, badResumptionToken, badVerb, cannotDisseminateFormat, idDoesNotExist, noRecordsMatch, noMetadataFormats, noSetHierarchy.

- ***Selective Harvesting:*** By selective harvesting the service provider issues requests to harvest a selective portion of the repository. There are two selective harvesting criteria – datestamps and setmembership.

***What are the OAI-PMH Verbs***

Here '*verb*' means request type which the service provider/harvester sends to get responses from data providers. There is a standard set of 6 verbs:

- Identify
- ListMetadataFormats
- ListSets
- GetRecord
- ListIdentifiers
- ListRecords

| | **Function** |
|---|---|
| Identify | Description of repository |
| ListMetadataFormats | Metadata format supported by the repository |
| ListSets | Sets defined by repository |
| ListIdentifiers | Retrieves unique identifiers of the item |
| ListRecords | Used to harvest records from the repository |
| GetRecords | Retrieves individual metadata record from the repository |

***Protocol Requests and Responses (OAI – PMH version 2.0)***

**Identify:** This verb is used to retrieve information about the repository. In response the data provider can send information on elements like *repository name*, *base URL* of the repository, *OAI-PMH version*, *Granularity*, etc. supported by the repository, etc.

Example
OAI-PMH Request:

http://drtc.isibang.ac.in/oai/?verb=Identify
OAI-PMH Response

*Figure1: An OAI-PMH argument*

Bad argument (e.g., http://drtc.isibang.ac.in/oai/?verb=identify) may lead to following error response, as the verbs are case sensitive:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2005-02-26T07:15:23Z</responseDate>
  <request>http://drtc.isibang.ac.in/oai/</request>
        <error code="badVerb">Illegal verb</error>
  </OAI-PMH>
```

**ListMetadataFormats:** This verb is used to retrieve the metadata format used by the repository.

Example
OAI-PMH Request:
http://drtc.isibang.ac.in/oai/?verb=ListMetadataFormats

OAI-PMH Response

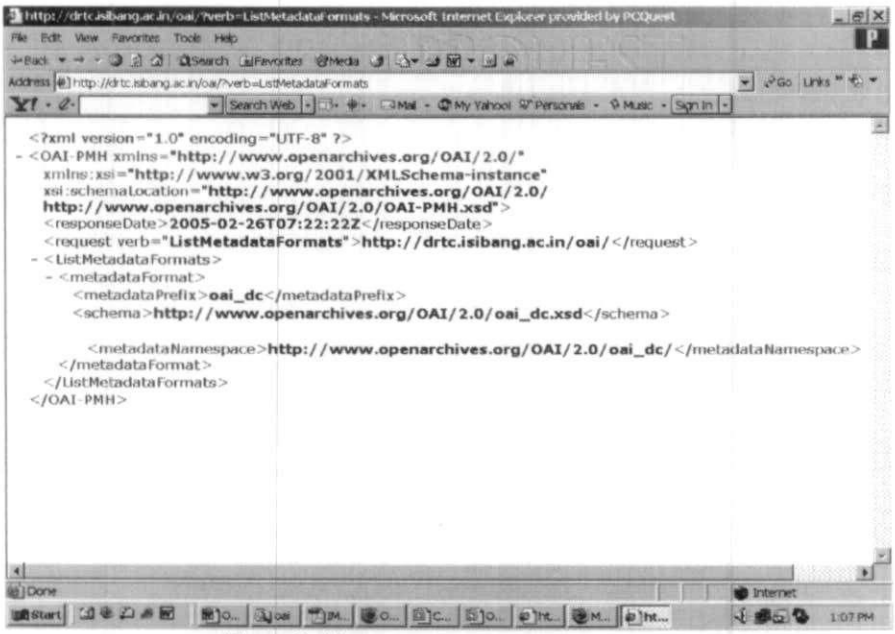


*Figure2: List sets*

**ListSets:** The verb is used to retrieve the set structure in which the metadata may be organized (hierarchical or flat) in the repository. The service provider can harvest metadata of a particular set of the repository according to the area of interest. For e.g. the harvester which collecting metadata of Library and Information Science (and in that only journal articles and theses/dissertations) only will specify the set in a repository which covers different subject resources.

Example
OAI-PMH Request:
http://drtc.isibang.ac.in/oai/?verb=ListSets

OAI-PMH Response


*Figure3: ListIdentifiers*

**ListIdentifiers:** This verb is used for selective harvesting of headers only not full record of the resources. Header includes the unique identifiers and properties (like date stamp, status attribute, metadataPrefix, resumptionToken, etc.) to identify the item.

Example
OAI-PMH Request:
http://drtc.isibang.ac.in/oai/?verb=ListIdentifiers&metadataPrefix=oai_dc

OAI-PMH Response

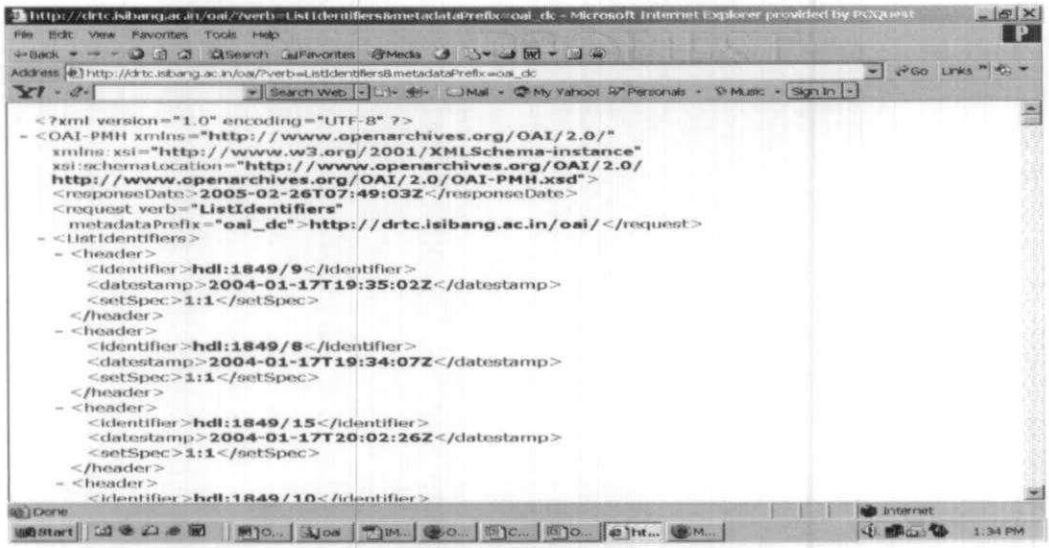


*Figure4:List Records*

**ListRecords:** Use to retrieve metadata records for multiple items. Parameters used are: starting date, ending date, set (set to harvest from), resumptionToken (for flow control mechanism), metadataPrefix (for metadata format)

Example
OAI-PMH Request:
http://drtc.isibang.ac.in/oai/?verb=ListRecords&metadataPrefix=oai_dc&from=2002-12-01

OAI-PMH Response

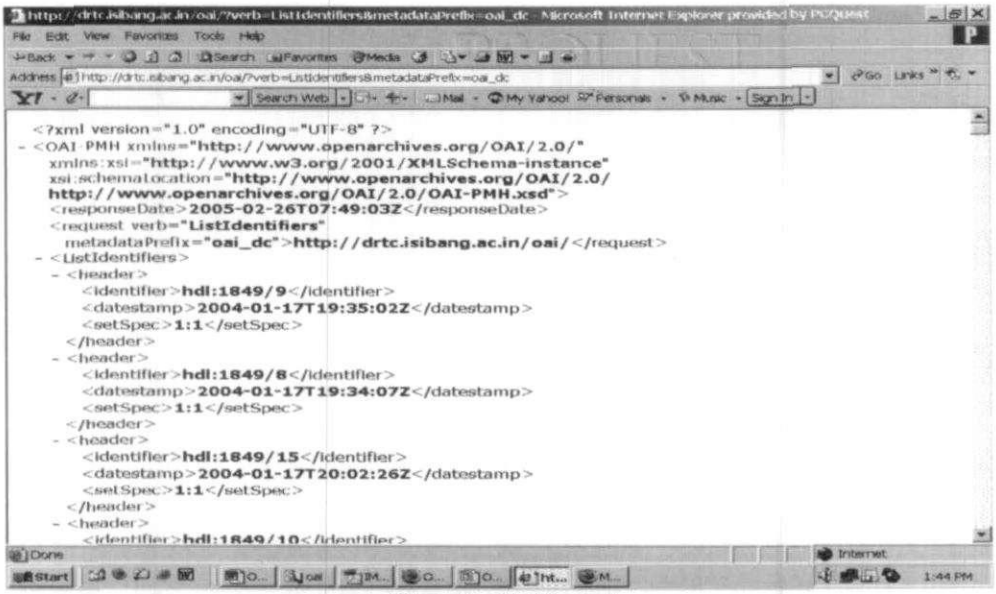


*Figure5: GetRecord*

**GetRecord:** Use to retrieve metadata record for one or specific item only. The required parameters are: Identifier (unique identifier for the item) and metadataPrefix (for metadata format)

Example
OAI-PMH
Request:http://drtc.isibang.ac.in/oai/verb=GetRecord&identifier=hdl:1849/99&metadataPrefix=oai_dc

OAI-PMH Response

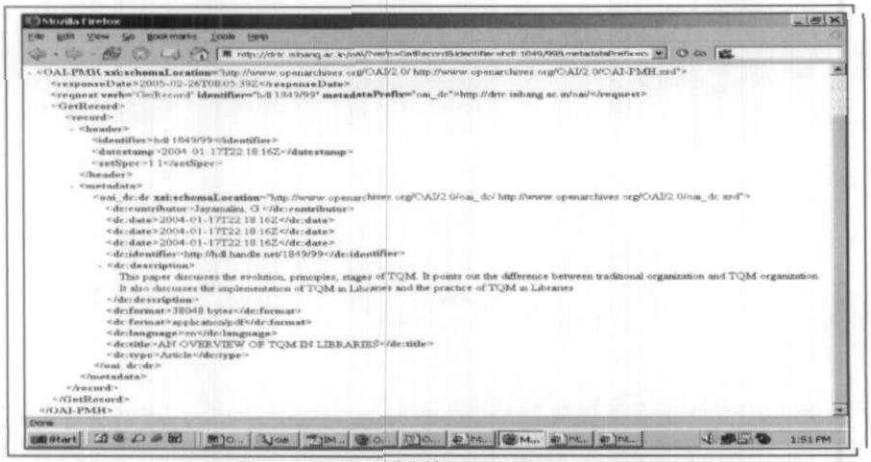

*Figure6: OAI-PMH response*

**Major Software supporting OAI-PMH (Open Archives Forum)**

**OAI Harvester Software**

- Arc (*http://arc.cs.odu.edu/*)
- Citebase (*http://citebase.eprints.org/cgi-bin/search*)
- CYCLADES (*http://www.ercim.org/cyclades/*)
- DP9 (*http://arc.cs.odu.edu:8080/dp9/index.jsp*)
- MeIND (*http://www.meind.de/*)
- METALIS (*http://metalis.cilea.it/*)
- my.OAI (*http://www.myoai.com*)
- NCSTRL (*http://www.ncstrl.org/*)
- Purseus (http://www.perseus.tufts.edu/cgi-bin/vor)
- Public Knowledge Project – Open Archives Harvester (http://pkp.ubc.ca/harvester/)
- OAICAT (*http://www.oclc.org/research/software/oai/cat.htm*)
- OAI Repository Explorer (*http://re.cs.uct.ac.za/*)
- OAIster (*http://oaister.umdl.umich.edu/o/oaister/*)
- OASIC (Open Archvies en SIC) (*http://oasic.ccsd.cnrs.fr/*)
- OAIHarvester (*http://www.oclc.org/research/software/oai/harvester.htm*)
- DLESE OAI Software (http://dlese.org/oai/index.jsp)

**Future Prospects**

Some more work has to be done in order to make OAI-PMH as a complete globally accepted metadata harvesting protocol:

- Tools and software has to be developed by which the non-OAI-PMH compliant repositories can be converted into OAI-PMH compliant so that the repository can be made data provider.
- The higher versions of the protocol should be made compatible of the lower ones.

At metadata creation level some standardization is required, as a particular resource is described inconsistently at different repositories. Vocabulary control measures should be also taken care of.

Still some more improvements are awaited in OAI-PMH protocol, and then only we can ensure a comprehensive view of the resources available on a particular subject to our end-users.

**References**

Caplan, Priscilla. Metadata Fundamentals for All Librarians. American Library Association Editions. 2003.
The Open Archives Initiative Protocol for Metadata Harvesting Protocol Version 2.0 http://www.openarchives.org/OAI/openarchivesprotocol.html#Item)
Rhyno, Art. Using Open Source Systems for Digital Libraries. Westport: Libraries Unlimited. 2004, pp 26.
The Open Archives Initiatives Frequently Asked Questions. Protocol Version 2.0 of 2002-06-14 *http://www.openarchives.org/documents/FAQ.html*
Open Archives Forum: Information Resource Database
http://www.oaforum.org/oaf_db/list_db/list_software.php#top)

# Configuration of DSpace with Apache and Tomcat

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

DSpace is written Java and JSP (Java Server Pages). For JSP to run either Tomcat or Jetty or Resin or Jboss is required. However, as Tomcat is the most preferred choice, this notes deals with Tomcat.

DSpace configuration can basically be in three ways:

- DSpace + Tomcat
- DSpace + Tomcat + SSL (Not covered here)
- DSpace + Tomcat + Apache
- DSpace + Tomcat + Apache + SSL

## DSpace + Tomcat

This is the most simple configuration. But the disadvantage is that the DSpace URL will have the port number, which is not the normal practice, when you inform the world about your repository site. In addition, apache is very robust and more secure environment. It is not recommended, as the URL would like

http://xyz.ac.in:8080/xmlui or
http://xyz.ac.in:8080/jspui

## DSpace + Tomcat + Apache

This is the most preferred choice. The LibLiveCD is configured using this method, whereas LDL (Librarian's Digital Library https://drtc.isibang.ac.in ) is configured with the addition of SSL. To integrate Tomcat with Apache we require a connector. In this notes, we use *mod_jk* connector. Once this is configured any request to the repository site first will go to Apache and Apache will pass on the request to Tomcat and when Tomcat responds to Apache with some content, Apache will pass on the content to the end user. The URL of the repository will look like the following without the port number.

http://xyz.ac.in/xmlui or
http://xyz.ac.in/jspui

Setting up DSpace + Tomcat + Apache: In this case, we require apache2 and mod-jk packages and we need to modify two files *default* and *workers.properties*

- **Step 1:** Install Apache and Mod-jk: You require apache2 server and mod-jk connector. In Ubuntu or Debian, you may use the following command to install both the software.

```
apt-get install apache2 libapache2-mod-jk
```

- **Step 2:** Modify default: Edit the file */etc/apache2/sites-available/default* and insert the following lines in *<VirtualHost>* directive and also after the end tag *</VirtualHost>*

Add the following lines in VirtualHost before ending VirtualHost

```
JkMount /xmlui  dspace_worker

JkMount /xmlui/* dspace_worker

JkMount /jspui  dspace_worker

JkMount /jspui/* dspace_worker

JkMount /oai    dspace_worker

JkMount /oai/*  dspace_worker

JkMount /solr   dspace_worker

JkMount /solr/* dspace_worker

JkMount /lni    dspace_worker

JkMount /lni/*  dspace_worker

JkMount /sword/* dspace_worker

JkMount /sword  dspace_worker


</VirtualHost>
```

- **Step 3:** Modify workers.properties: Edit the file */etc/libapache2-mod-jk/workers.properties*, so that the following parameters are set correctly. Following lines give a sample, however, depending on the directories of Java and DSpace, you may according modify the path names. And also make sure your port numbers are correct.

  workers.tomcat_home=/home/dspace/apache-tomcat-7.0.6/

  workers.java_home=/usr/lib/jvm/java-6-openjdk

  worker.list=dspace_worker
  worker.dspace_worker.port=8009

  worker.dspace_worker.host=localhost

  worker.dspace_worker.type=ajp13
  worker.dspace_worker.lbfactor=1

  worker.loadbalancer.balance_workers=dspace_worker
- **Step 4:** Restart apache.

# Using Gmail in DSpace

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

One of the greatest advantages of using Linux servers is that they come with three mail server software – sendmail, postfix and exim4. In addition, the end user can have a GUI client (Evolution, Thunderbird) or web based mail client (Squirrelmail) to access and send mail. For the mail server to pass the mail to a client you require an IMAP (Internet Mail Access Protocol) server or POP3 (Post Office Protocol). Linux offers a number of IMAP and POP3 servers. The LibLiveCD comes with exim4 mail server, dovecot IMAP server and Squirrelmail.

Even if an organization does not have a mail server, if it has a public IP number and address, LibLiveCD can be used to send and receive mail. If an organization does not have a public IP address, they can still use exim4 mail server of the LibLiveCD and make DSpace send mails. In other words, if one does not have a mail server, they can still use exim4 and make DSpace send mail to its members, albeit with some limitations. The rationale behind it is, if one lives in Railway station or Bus stand he can still write letters to his friends, though he cannot receive any postal mail.

## Using Gmail

This presentation deals with the worst case scenario, i.e. one has DSpace installed successfully, but could not configure a mail server for various reasons. In such a case one can use gmail to send mails generated by DSpace.

The following notes is taken from (Lewis, 2009):
http://blog.stuartlewis.com/2009/09/05/using-gmail-with-dspace/

- **Step 1**: Create a new Gmail account. (a new Gmail account may be created for this purpose or use yours if you have already one, if you prefer)

- **Step 2:** open dspace.cfg file in an editor and fill up the information for *mail.server.username* and *mail.server.password*. You may leave the rest of the parameters.

```
# SMTP mail server
mail.server=smtp.gmail.com

# SMTP mail server authentication username and
password (if required)
mail.server.username = your-user-name@gmail.com
mail.server.password = your-gmail-password

# Pass extra settings to the Java mail library.
Comma separated, equals sign between
# the key and the value.
mail.extraproperties =
mail.smtp.socketFactory.port=465, \
mail.smtp.socketFactory.class=javax.net.ssl.SSLSoc
ketFactory, \
mail.smtp.socketFactory.fallback=false
```

**Further Readings**

Lewis, Stuart. (2009, September 5). *Using Gmail with DSpace.* Retrieved from http://blog.stuartlewis.com/2009/09/05/using-gmail-with-dspace/
DSpace. *How to use Gmail with DSpace.* Retrieved from http://drtc.isibang.ac.in/tips

# Changing DSpace Themes

**Devika P. Madalli**

Documentation Research and Training Centre
Indian Statistical Institute
Bangalore
*devika@drtc.isibang.ac.in*

## Introduction

The very purpose of introducing Manakin is to encourage developers and even users having knowledge of XML, CSS and related topics to develop themes. Hence, the move over to xmlui leaving behind jspui. Presently, only a few themes are developed. They are:

- Default
- Classic
- Kubrick
- Mirage


*Figure1: The Default Manakin Theme*

## The customisation

- **Step 1:** To change the theme of xmlui, edit the file

```
/dspace/config/xmlui.xconf
```

- **Step 2:** Go to <themes> tag, which should look like the following

```
<themes>
        <!-- <theme name="Default Reference
Theme" regex=".*" path="Reference/" /> -->
        <!-- <theme name="Default Kubrick
Theme" regex=".*" path="Kubrick/" /> -->
        <!-- <theme name="Atmire Mirage Theme"
regex=".*" path="Mirage/" /> -->
        <!-- <theme name="Classic" regex=".*"
path="Classic/" /> -->
</themes>
```

- **Step 3**: Uncomment the line to choose a different theme. In XML the comment lines begin with

```
"<!--: and end with "-->"
```

For example, if you wish to choose, Kubrick theme, change the line

```
<!-- <theme name="Default Kubrick Theme"
regex=".*" path="Kubrick/" /> -->
```

**To**

```
<theme name="Default Kubrick Theme" regex=".*"
path="Kubrick/" />
```

**Note:** Make sure the other lines in the tag *<themes>* are commented.

- **Step 4:** Stop and start tomcat
- **Step 5:** In the browser, enter the URL having xmlui

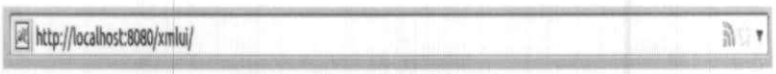


*Figure2:xmlui url*

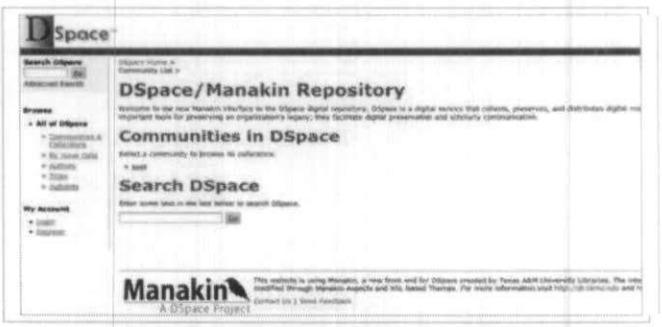


*Figure 3: Kubrik Theme*

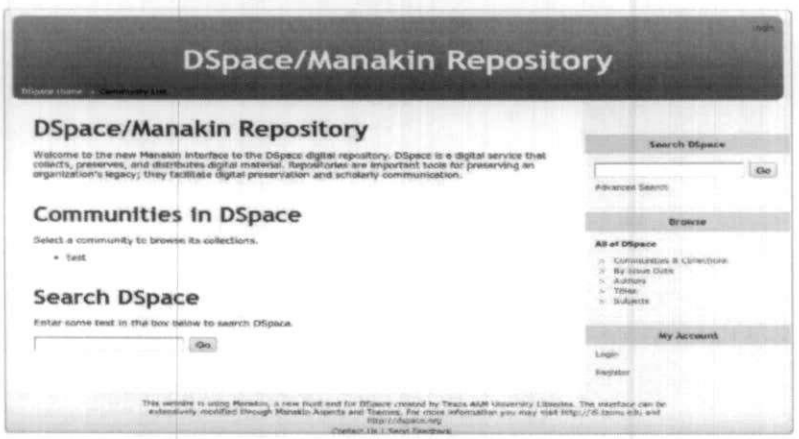


*Figure 4: Classic Theme*


*Figure5:Mirage Theme*

## Conclusion

In Step 2, a total of 4 themes are presented. The default Manakin theme name is "Default Reference Theme". If you do not make a choice or uncomment the other, DSpace uses plain Manakin theme i.e. "Default Reference Theme". The theme having the name "Classic", presents a look alike of jspui interface, but developed using Manakin. The theme Kubrick is yet another interface developed using Manakin.

Compared to the earlier Manakin, Mirage, which is introduced in the latest version of DSpace 1.7.0 claims easier customization and better performance. In the screenshots, except the Mirage, the other three themes are developed using Manakin. In that sense, Manakin and Mirage are emerging as competitors. Perhaps the future version of DSpace, may even drop jspui interface. Notes on developing themes using Mirage can be found at https://wiki.duraspace.org/display/DSDOC/Mirage+Configuration+and+Customization

# Upgrading DSpace for the Impatient

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

Periodically we come across newer versions of DSpace as it has a very active developer community. To keep your repository up-to-date with the newest releases of DSpace, this note explains an unorthodox (and unofficial) yet simple way of upgrading DSpace. Often members of DLRG (http://drtc.isibang.ac.in/dlrg), keep asking how to upgrade to the newer version of DSpace.
The most important files/directories with regard to upgrading are: *assetstore* directory and the *database files*.

- **Step 1:** Take a back up of the entire dspace directory (Not essential, but to be on guard).
  We require only the *assetstore* directory, where the full-text or media files are stored
- **Step 2:** Take a back up of the dspace database, using the following command, as dspace user:

```
pg_dump > backup-17-Feb-2011
```

- **Step 3:** Ignoring the earlier instance of dspace, create a new dspace instance, say "newdspace", perhaps in the directory called */newdspace'*. Login as superuser, using some command like:

```
# adduser --home /newdspace
```

  This creates a new user called newdspace in the directory */newdspace*

- **Step 4:** create in postgres a new database user called newdspace
- **Step 5:** create database called newdspace as newdspace user
- **Step 6:** compile newdspace using maven and ant.

NOTE: before compiling, modify the *dspace.cfg* file in

```
/dspace-1.7.0-release/dspace/config
```

Especially modify the following parameters

```
dspace.dir=/newdspace
db.username = newdspace
db.password = dspace
```

- **Step 7:** Drop database newdspace, as this will be filled with new database structure, after you run ant fresh_install
- **Step 8:** Again create database with same name i.e. newdspace. This is to create an empty database.
- **Step 9:** Fill up the newdspace database with old data and its structure. That is, with the data which was backed up in Step 2.
  Ex: psql < backup-17-Feb-2011
- **Step 10:** Now go to the directory *'/dspace-1.7.0-release/dspace/etc'* in DSpace source files you will find the following files:

```
database_schema_11-12.sql
database_schema_12-13.sql
database_schema_13-14.sql
database_schema_14-15.sql
database_schema_15-16.sql
database_schema_16-17.sql
```

- **Step 11:** Apply each file in a sequence from the version of your old dspace instance to the latest, one after another. For example, your old version of Dspace is dspace-1.2.1, run the following commands one after another

```
$ psql <  database_schema_13-14.sql
$ psql < database_schema_14-15.sql
$ psql < database_schema_15-16.sql
$ psql < database_schema_16-17.sql
```

As DSpace major versions change the database tables and structure, the above commands will change the older data into new structure.

- **Step 12:** Copy all the files assetstore directory of the old version into asset store directory of newdspace. The command may look like the following

```
$ cp -r /dspace/assetstore/*/newdspace/assetstore
```

- **Step 13:** Stop the tomcat of the old instance of DSpace repository and run the tomcat for the newer version of DSpace.
- **Step 14:** Once you are satisfied with the new instance for a month, you can destroy the old one.

# DSpace LiveCD

**ARD Prasad**

Documentation Research and Training Centre
Indian Statistical Institute
*Bangalore*
*ard@drtc.isibang.ac.in*

## Introduction

A LiveCD is a bootable CD. LiveCD normally comes with a variant of Linux. If the system is booted with livecd, it will not be loaded on your hard disk. In other words, the hard disk will not be touched. You work only from the CD. What ever you do, they will disappear once you shut down. Nothing will be saved, as CD can not be written and you can resume normal operations with your system

One can have a livecd on:

CD (of 700MB) or DVD or USB memory stick (fastest) or Virtual Machines like VBOX, qemu, VMWare, kvm, etc. (slowest – unless you have more than 2GB main memory)

**Pupose:** One can test an operating system without actually loading it and even test all the applications under an OS that come with the livecd without loading them. If one is satisfied, one can load the OS onto the hard disk.

This release of LibLiveCD is created on 5th July, 2011 and configured using Ubuntu 10.04.1 (Lucid: LTS - Long Term Support version) and DSpace 1.7.2

## How to download liblivecd

- Go to: http://sf.net/projects/liblivecd OR *http://liblivecd.sf.nt*. Download the latest ISO file. Also download liblivecd.iso.md.
- After downloading ISO file run:

  `md5sum liblivecd.iso` (generates a unique number)

- See the number in liblivecd.iso.md5 and the number md5sum has generated are same

  You can burn a CD OR prepare a bootable memory stick, If you are using Ubuntu (faster compared to CD)

  *System → Administration → USB Startup Disk Creator*

- Use qemu or kvm (requires more main memory)

  It could be very slow/fast depending on your Main Memory size. Run the following command

  ```
  # qemu -cdrom new/remaster-new-files/liblivecd.iso -boot
  d -m 512
  ```

  You may replace kvm with qemu

- Make sure your CD is of 700MB capacity
- You should **NOT** copy the ISO image to CD
- You should **BURN** ISO image to CD
- After burning, check whether CD has directories like

  *casper, dists, install, isolinux, pool, preseed etc.*

- Before booting choose CD-ROM as the first boot device in CMOS setup.

**How to install on har disk**

Install liblivecd on a fresh system, where there is no data.If you already have another OS anddata, make sure you have an empty partition. Many Linuxes allow dual boot, though not recommended for a server

**Caution**: You should be careful with disk partition, if you have already some data on your hard disk

**DSpace**

DSpace Classic Interface (*http://localhost/jspui*)
DSpace XML Interface (*http://localhost/xmlui*)
Dspace Admin login: `dspace@localhost`
Password: `library`

DSpace BaseURL: *http://localhost/oai/request*
DSpace OAI Identify
DSpace OAI ListRecords

**NOTE**: As tomcat is integrated with apache you do not have to use port number like 8080. Another advantage is: even if your firewall blocks 8080 port, your DSpace can be accesed through port 80 of apache. LiveCD can even send mails, if you are running on a machine with Internet IP number and not behind firewalls

**Web Mail Client**
**Squirrelmail**
You may try:

Login: `dspace`
password: `library`

**NOTE**: This mail client uses exim4 mail server and gets mail through dovecoat IMAP server. Dovecat can facilitate accessing your mail using any of the mail clients like evolution, thunderbird, outlook express etc. If your organization has a firewall, read Further Configuration and talk to your network administrators.

**Logins and Passwords for shell, mysql etc.**

password for dspace: `library`
password for koha: `library`
Login for mysql admin user: `root`
Password for mysql admin user: `library`

***LiveCD: What sofware is included and excluded***

Please send feed back to discusssion forum

*http:drtc.isibang.ac.in/dlrg* OR to me: `ard@drtc.isibang.ac.in`
If you are already a member, write to:
`dlrg@drtc.isibang.ac.in`

This LiveCD comes with pre-configured Ubuntu-10.04.1 (Lucid: LTS - Long Term Support version) for i386 machines (Not a 64bit version). To accommodate many server software and still keep it below 700MB, many desktop related software are not installed. However, one can install them using 'synaptic'.

- **Software included**

  Minimal Gnome Desktop
  Apache2
  Open-ssh server
  vsftp server
  PostgreSQL
  Exim4 (mail server)
  dovecot (IMAP server)

SquirrelMail (web based mail client)
OpenJDK Java and Tomcat
FoxitReader (to read PDF files)

- **Sofware excluded**

  Open Office (instead abiword, gnumeric are provided )
  evince
  mail clients
  languages other than English
  games
  Multimedia related s/w
  and Many More ...

**How to add addtional software**

However, if you wish to add any software go to
*System->Administration->Synaptic*

**Further Information**

To access any one of the pre-configured library services you may use logins and passwords given in the write up As this is a LiveCD, it would not touch your hard disk. Once you come out of the LiveCD. You can resume your normal operations on your system. However, if you wish install the contents of this CD, you may click the icon "Install Ubuntu 10.04.2". You only have to be careful with Partitioning the disk, as it allows multi-OS boot"

One disadvantage with any LiveCD is, you can/should NOT install it on an existing server on which you already have lots of data and software configured. In future, '.deb' files may be released to facilitate configuring theses software on Debian Linux compatible operating system like Ubuntu

***Further Configuration***

Once the LiveCD is installed on to your hard disk, you may further modify the configuration

General

Depending on whether you wish to use the newly installed system on LAN or Internet, modify the following files.

- Press Alt+F2 and enter: sudo gedit /etc/hosts: Enter the IP Address and Fully Qualified Domain Name
- Press Alt+F2 and enter: sudo gedit /etc/hostname: Enter the hostname of your system without domain name Press Alt+F2 and enter: sudo hostname
- On the LiveCD exim4+dovecot+squirrel mail are pre-installed. However, to reconfigure the mail server
- Press Alt+F2 and enter: sudo dpkg-reconfigure exim4-config
  Fill the information in the screens that appear

Alternatively, instead of using Alt+F2, you may open a terminal using the menus: *Applications -> Accessories -> Terminal* and use sudo commands

**DSpace**

For extensive customization of the configuration go to /home/dspace/config/dspace.cfg. All the parameters are provided with help information in this file.If you wish to use your organizations e-mail server change the parameter "mail.server".

Whenever you modify dspace.cfg file, do not forget to Stop and Start Tomcat. To do that used the commands:

- Press Alt+F2 and enter: sudo /etc/init.d/dspaced stop
- Press Alt+F2 and enter: sudo /etc/init.d/dspaced start

Your organization may have firewalls, talk to your system administration to ensure mails are sent from your system.

**DSpace Documentation**

For more information go to DSpace site *(www.dspace.org)*

# Regional e-Repository Workshop on DSpace software

*27-30 July 2011*

| | Name | Address | E.mail |
|---|---|---|---|
| **Resource Persons** | | | |
| 01 | Dr. Medalli Devika Pandurang | Documentation Research & Training Centre (DRTC)<br>Indian Statistical Institute (ISI)<br>8th Mile Mysore Road<br>Bangalore 560 059<br>**India** | devika@drtc.isibang.ac.in |
| 02 | Prof. Areti Ramachandra Durga Prasad | Documentation Research & Training Centre (DRTC)<br>Indian Statistical Institute (ISI)<br>8th Mile Mysore Road,<br>Bangalore 560 059<br>**India** | ardprasad@gmail.com |
| **List of Participants** | | | |
| 01 | Mr. Sheydaee, M | R&D Manager<br>Iranian Research Institute for Information Science and technology<br>Teharan, Enghelab AVE<br>No;1188<br>**Iran** | sheydaee@irandoc.ac.ir |
| 02 | Mr. Mohd Ikhmil Firdausz Bin Mohd Hanif | National Library of Malaysia<br>Information Technology Division<br>232, Jalan Tun razak<br>50572 Kuala Lumpur<br>**Malaysia** | firdausz@pnm.my |
| 03 | Mr. Amorn Hovsomboon | National Electronic and Computer Technology Centre (NECTEC)<br>Nat. Science & Technology Development Agency<br>112, Thailand Science Park<br>Phahougothin Road<br>Klong 1 , Klong luang<br>Pathumathani 12120<br>**Thailand** | amorn@nectec.or.th |

| | | | |
|---|---|---|---|
| 04 | Mr. Heng Sovannarith | Hun Sen Library,<br>Royal University of Phnoro Penh<br>24 Eo, Street 578<br>Sangkat Boeng kak II<br>Toulkork<br>Phrom Penh<br>**Cambodia** | heng_sovannarith@yahoo.com |
| 05 | Mr. Sophal, Sok | Mm Khlod Vibolla/Director<br>National Library of Cambodia<br># 92, St Daun Penh<br>Phnom Phenh<br>**Cambodia** | soksophal1@gmail.com |
| 06 | Mr. Yunus Khomaeni | Act Head/Infrastructure & Supra - Structure of Data<br>Data and Information Division<br>The Ministry of Research and Technology<br>Republic of Indonesia<br>Jl. MH Thamrin no 8<br>J BPPT Bld 11, 5th floor<br>Jakarta 10340<br>**Indonesia** | yunus@ristek.go.id |
| 07 | Mr. Purbey Jagdish Kumar | Ministry of Education<br>Nepal National Library<br>Harihar<br>Bhawan<br>Pulchowk,<br>**Nepal** | jkpurbey@yahoo.com |
| 08 | Ms. Do Nhu Tho | Digitalization Division<br>National Agency for Science & Technology Inforjmation<br>NASATI<br>24Ly Thuong Kiet Street,<br>Hoan Kiem<br>Hanoi,<br>**Vietnam** | donhutho2812@gmail.com |

| | | | |
|---|---|---|---|
| 09 | Ms. Souphaphone Phouyavong | Technical Officer<br>Department of Information, NAST<br>Prime Ministers office<br>Sisrath Village,<br>Chanthabouty district,<br>Vientiane<br>P.O.Box 2279<br>**Lao PDR** | psouphapone@yahoo.com |
| 10 | Ms. Annisa Pranowa | The Ministry of Research & Technology Republic of Indonesia<br>Jl. MH Thamrin no 8<br>J BPPT Bld 11, 5th floor<br>Jakarta 10340<br>**Indonesia** | nisa@risteck.go.id |
| 11 | Ms. Harshini Dissanayaka | Assistant Librarian<br>University of Peradeniya<br>Peradeniya<br>**Sri Lanka** | harshani.dissanayake@gmail.com |
| 12 | Mr. C.A.B.Wickramasinghe | Library Assistant<br>National Science Foundation<br>47/5,Vidya Mawatha<br>Colombo -07<br>**Sri Lanka** | chanaka@nsf.ac.lk |
| 13 | Mr. I.D.Kusala Lakmal Fernando | Asst. Librarian/Main Library<br>University of Ruhuna<br>Wellamadama<br>Matara<br>**Sri Lanka** | kusala@lib.ruh.ac.lk |