

# DSpace Architectural Review

## Document details

Project: SHERPA DP  
Work Package: Kirti Bodhmage  
Version: Version 1.0  
Document date: 06/09/2005  
Change history:

<i>Date</i>	<i>Version</i>	<i>Author</i>
06/09/2005	First version	Kirti Bodhmage

---

## Contents

Summary .....	2
Introduction.....	2
Data Model .....	2
Export Interfaces in DSpace: .....	3
1. ItemExport Built-In class.....	3
2. DSpace Migrate Utility.....	4
3. METSExport class .....	4
4. Lightweight Interface for DSpace .....	4
Similar Projects: .....	5
1. Web Services for DSpace .....	5
2. CWspace .....	5
Potential Approaches: .....	5
1. Implement Web Service Interface .....	5
2. Using Lightweight Interface for DSpace.....	6
3. Secure Copy Protocol.....	6
4. Download, Upload Servlets .....	6
References: .....	6

## Summary

This document investigates the data model and export interfaces used by DSpace. The primary objective is to identify data migration mechanisms that may be suitable for the Eprint Archive-to-AHDS (Preservation Service) data transfer. The options outlined are assessed according to the functionality they currently offer.

## Introduction

DSpace is a digital repository system that captures, stores, indexes, preserves, and redistributes an organization's research material in digital formats. The software has been co-developed by MIT and Hewlett Packard for use on Windows and Unix/Linux platforms and is distributed under the BSD open source licence. The software is based upon the PostgreSQL database and is written in Java.

## Data Model

The DSpace data model contains multiple layers and is derived, at least to a limited extent, from a library-based structure. The model is divided into Communities, which typically correspond to a laboratory, research centre or department. The Community is further divided into collections of items, which in turn, are composed of bundles of bitstream information. Bitstreams that are somehow closely related, for example HTML files and images that compose a single web document are organised into bundles. Handles are assigned to communities, collections, and items. Bundles and bitstreams are not assigned a Handle.

The organisational model is intended to be flexible and can be changed as and when necessary.

The following example illustrates the DSpace Data Model.

Community	Art and Humanity Department
Collection	History Collection
Item	Reports on World war II
Bundle	PDF documents, HTML Files
Bitstreams	Single PDF file
Bitstream Format	Adobe Portable Document Format

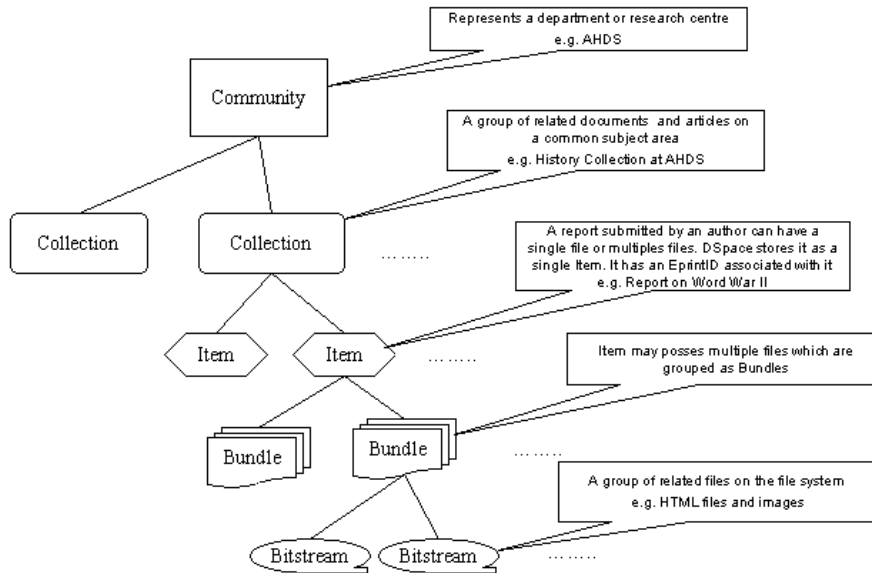


Figure 1: The DSpace data model

The DSpace workflow is designed to simplify the submission process and gather information relevant to the deposited data. The depositor is requested to complete a series of online forms containing relevant fields, such as title, author, etc.

Once complete, the metadata is written to the DSpace Workflow Manager writes the information to the relevant database tables and copies the bitstream – the eprint to the local file system. A unique Handle identifier is allocated to each submission, to maintain the relationship between bitstream(s) and metadata.

## Export Interfaces in DSpace:

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

### 1. ItemExport Built-In class

DSpace has a class ItemExport in its application layer that has following functions:

- WriteMetadata - writes the Dublin Core metadata to a file in specified directory
- WriteBitStream - writes bitstream, actual file to a directory
- WriteHandle - creates the file 'handle' which contains the handle, a persistent identifier assigned to the item

These functions write a Dublin core record, licence file and the content file into a specific directory.

The syntax to call this utility is:

```
dsrun org.dspace.app.itemexport.ItemExport --type=COLLECTION --id=collID --
dest=dest_dir --number=seq_num
```

After executing the above command output is as follows:

```
archive_directory/
item_000/
  dublin_core.xml -- qualified Dublin Core metadata
  contents       -- text file containing one line per filename
  file_1.doc     -- files to be added as bitstreams to the item
  file_2.pdf
```

```
item_001/  
  dublin_core.xml  
  contents  
  file_1.png  
  ...
```

Here item\_000 is the actual item in repository.

**Comment:**

An institutional repository can modify the existing interface to extract the desired data from the file system and tables. It may be required to add few extra columns in the database to flag the new modification in the eprints. This interface does not provide any networking solution for the data migration. The Push or Pull methodology has to be implemented separately.

## 2. DSpace Migrate Utility

The DSpace migration utility migrates the data from one DSpace instance to another by retaining the directory structure.

**Comment:**

This utility is incomplete and there is no plan in the developer community to develop it further.

## 3. METSExport class

The METS-based export tool exports DSpace items to a file system, accompanied by METS compliant metadata. This tool is currently under development by MIT and HP. The METS Flocat element provides a pointer to the location of a content file in an item in the form of URL. These URLs can be used to retrieve the bitstreams. The general pattern suggested by MIT for migrating the content using this tool is OAI-PMH or WebDAV web services, which is under development at MIT.

**Comment:**

As AHDS plans to store metadata in the METS format, this utility will be useful to extract the Metadata in METS format from an Institutional repository. An institutional repository needs to patch the existing DSpace code with the new classes and update the DSpace installation. There are two ways to change the Metadata included in METS, either by changing the METSExport file or changing the dc2mods.cfg file.

A Plug and Play version of this utility is available at  
[https://sourceforge.net/tracker/index.php?func=detail&aid=1244813&group\\_id=19984&atid=319984](https://sourceforge.net/tracker/index.php?func=detail&aid=1244813&group_id=19984&atid=319984)

Developer: Robert Tansley [robert.tansley@hp.com]

## 4. Lightweight Interface for DSpace

This interface aims to provide a lightweight, generalised, extensible framework for the networked DSpace instances. This is based on the WebDAV, which stands for "Web-based Distributed Authoring and Versioning". It is a set of extensions to the HTTP protocol, which allows users to collaboratively edit and manage files on remote web servers. It has a set of APIs.

GET: The GET method can return an item as a package, e.g. a Zip file with a METS manifest. The GET method has to be called individually for each Item.  
PROPFIND: The PROPFIND method searches a collection, lists the collection and community and provides all metadata as WEBDAV properties.  
PUT: The PUT method gives the ability to upload resources to a server.

**Comments:**

This interface is under development at MIT. This is not exactly data extraction tool but it gives a networking interface to connect any of the systems with DSpace. This tool can be used in conjunction with either ItemExport or METSExport interface.

As no initial version of software is available, customization details cannot be given. The release date has not been confirmed by MIT.

Developer: Larry Stone [lcs@mit.edu]

Project URL: <http://wiki.dspace.org/LightweightNetworkInterface>

## Similar Projects:

### 1. Web Services for DSpace

The CARET project at Cambridge is making DSpace interoperable with other systems (e.g. Course Management System) using Web services implemented in SOAP/WSDL in DSpace. It consists of 6 services that support the process of collection and dissemination of material in the repository by web services. These are:

- Search in the Collection
- Upload Service
- Download Service
- Package Service
- Ingest Service
- Session Setup Service

The consumer of these services has to write a client using published WSDL.

Project URL: <http://wiki.dspace.org/WebServiceForDSpace>

### 2. CWspace

The CWspace project at MIT is aiming to develop an Open Courseware Learning object and also to archive and preserve it. Preservation will be implemented by extending functionality in DSpace.

MIT will develop Web Services and workflows to exchange valuable course material with extant course management systems, both commercial and locally developed.

A lightweight network interface (LNI) based on WEBDAV is being developed to provide remote access to DSpace. Another option of SOAP/WSDL web services is underway.

Content packaging will be handled by METS.

Project URL: <http://cwspace.mit.edu/>

## Potential Approaches:

Four potential approaches may be taken to transfer data between an institutional repository and AHDS.

### 1. Implement Web Service Interface

DSpace Repository implements the web service interface to send and receive the metadata and contents. Send service exports the data from database and file system using ItemExport or METSExport Interface, packages it, compresses it, encodes it and sends to preservation system. Consumer, preservation system consumes the published services. Data transfer can be requested by preservation as and when needed. Technology alternatives:

- Axis/Soap, WSDL (Server side IR and Client side is Preservation)
- SAAJ APIs (SOAP with Attachments API for Java)
- SOAP RPCs and Servlet

### Comment:

Similar work is under development by the CARET project in Cambridge although nobody from CARET has responded regarding the project.

Being a web service based solution; it allows an institutional repository to communicate with different platforms and operating systems.

## 2. Using Lightweight Interface for DSpace

This is another web-based solution for network communication. An institutional repository has to implement the interface, which extracts the data using ItemExport or METSEXP interface, packages it and sends it across using this interface.

### Comment:

This solution will also have advantages of a web service interface. As this work is under development, details of the solution are not clear.

## 3. Secure Copy Protocol

SCP is a secure equivalent of FTP. Rsync or Scp can be used to synchronise the data on DSpace and Preservation system. A crontab for every night can be setup for the Unix based repositories, which will initiate the data transfer using SCP to the preservation system. ItemExport or METSEXP Interface can be used to extract the data.

### Comments:

The AHDS is using SCP to get the collection data from the Oxford University. This is the simple solution for the data transfer. Network traffic is key to the reliability and performance of the data transfer.

## 4. Download, Upload Servlets

Preservation implements a Data Upload facility that allows an institutional repository administrator to upload data to a specific directory on the preservation server. An institutional repository administrator runs the ItemExport or METSEXP tool and extracts the data from DSpace, zips it, encodes it and requests the file Upload to the preservation server.

### Comments:

Manual intervention required for uploading and downloading the eprints.

## References:

1. <http://wiki.dspace.org/PackagerPlugins>
2. <http://dspace.org/technology/system-docs/application.html>
3. <http://wiki.dspace.org/ChinaDigitalMuseumProject>
4. <http://wiki.dspace.org/NetworkInterfaces>
5. <http://www.caret.cam.ac.uk/> CARET project at Cambridge University] and MIT
6. <http://icampus.mit.edu/projects/DSpace.shtml> CWSpace project at MIT