

# Digital Preservation Exercise 2 - Repository System Comparison (CKAN and DSpace)

Gruppe-Option3-CKAN-DSpace3

Marc Dietrichstein - 0327606

Markus Neumeyer - 1225172

# Installation

## CKAN

The CKAN documentation lists the following installation options:

### Package

Installation as a debian (.deb) Package. Only supported on Ubuntu 16.04 and 14.04. The package installation only contains the CKAN software itself. All dependencies (Postgres, Solr, etc.) have to be installed and configured manually.

### Source

Compiling the project from source. The instructions assume that a debian based Linux distribution is used. Required dependencies have also to be installed manually. Community made guides to install CKAN from source on Windows exist, but can get tricky.

### Docker

CKAN contains docker compose configuration files which can be used to quickly set up a working CKAN installation and the relevant dependencies. The instructions still assume that the host is a Linux machine.

Since the Package and Source installation options require a Linux host and involve a lot of manual installation and configuration of the required dependencies we have decided to install CKAN via Docker Compose first. The host was running Mac OSX.

We installed it to a Windows machine as well with the help of mentioned community made guide, which was more time consuming but in the end led to the same stable result.

The manual steps while installing CKAN on windows include installing pgadmin, getting python to run on Windows like it does on Linux, which wasn't as easy at it first seemed to be, changing some setup files of CKAN as well as setting up a virtual machine which runs CKAN in the end.

The installation instructions worked for the most part, except for the database setup. This seems to be a known problem and the relevant issue is tracked [here](#).

The documentation itself seems to be a little bit out of date, when we had problems we were able to quickly find solutions on their issue tracker though.

## DSpace

DSpace is implemented as a collection of Java Web Applications (war) and has to be deployed via a Java Web Server like Tomcat, Jetty, etc.

The DSpace installation documentation provides two installation options: Binary and Source. Both variants require a maven build to be executed, which also requires the installation of a JDK, Ant and Maven.

The only runtime dependency is a database. The choice here is between PostgreSQL and Oracle.

We have decided to go with the source release. We were able to build the project without problems by following the instructions provided in the documentation. The final setup steps involved a lot of manual tasks like configuring the tomcat webapps to use, database setup, etc. A preconfigured docker option would have been useful.

## Content Organisation

### CKAN

#### Content Types

CKAN is written in python and supports all content types which are supported by python's [mimetypes](#) library. The definition of [custom content types](#) is supported via the CKAN plugin system.

#### Collections

CKAN Groups are a means to group datasets into collections.

#### Versioning and Identifiers

Versioning does not seem to be officially supported. Such functionality can be added by [installing a plugin](#) though.

CKAN also does not seem to support common identifiers systems like DOI. Such functionality can also be added by installing plugins, for example the [DOI plugin](#).

### DSpace

#### Content Types

DSpace manages Content Types in it's *Bitstream format registry* under *Registries* → *Format*. The most common Content Types are already preconfigured. Additional types can easily be added via the admin interface.

#### Collections

DSpace collections consist of a collection of Items which are the equivalent of DSpace's Datasets.

#### Versioning and Identifiers

Each item can optionally be provided with a number of different identifiers. The current list of possible identifiers consists of: ISSN, ISMN, Uri, ISBN, Gov't Doc # and Other.

Usage of DOI's is possible via the integration of [external identifier services](#).  
DSpace supports item level versioning, but this feature has to be [enabled explicitly](#).

## Metadata

### CKAN

CKAN uses a set of core metadata fields like tags, author, maintainer, etc., that should be useful for almost every upload.

Additional metadata can be added through an unlimited number of key/value-pairs, where the key (type of data) can be defined by the user itself, i.e.:

<b>Custom Field:</b>	Key: length	Value: 123
----------------------	-------------	------------

### DSpace

DSpace derived its metadata from the 15 basic Dublin-Core elements with the addition of DCTERMS.

DSpace allows you to add additional elements to the existing scheme through the definition of an element-name, qualifier and scope, as well as the addition of complete metadata schemes through their URI:

Add a new schema

Namespace: \*

Namespace should be an established URI location for the new schema.

Name: \*

Shorthand notation for the schema. This will be used to prefix a field's name (e.g. dc.element.qualifier). The name must be less than 32 characters and cannot include spaces, periods or underscores.

Add new schema

## Content Presentation

### CKAN

In CKAN the whole content of a data-set is listed with the name of each file and the description that is written while uploading the file.

Depending on the filetype, some files allow only a download (microsoft office files, music files), some allow a preview (images) and some allow to view the whole content directly via the UI (text-files).

#### Data and Resources

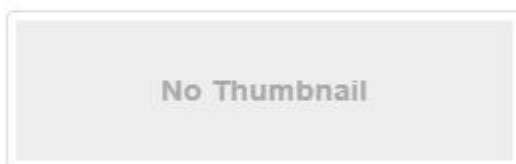
	<b>picture</b> picture	<a href="#">Explore ▾</a>
	<b>recipe</b> recipe	<a href="#">Explore ▾</a>
	<b>xml-file</b> xml-file	<a href="#">Explore ▾</a>

For example clicking on the “explore” button for the picture allows to choose “preview”, where a small preview of the picture can be seen. The xml-file can be viewed completely, while the recipe (a .docx-file) can only be downloaded.

## DSpace

In DSpace every collection (of files/submission) has a list of submissions, that can be sorted in different ways. When opening a single submission (which can contain one or more files), a thumbnail is shown as well as a short description of the submission and the list of files included with some additional metadata:

### test files - image



#### View/Open

-  anonymous picture (2.725Mb)
-  xml data (1.479Kb)
-  recipe (13.68Kb)

#### Date

2018-05-26

#### Author

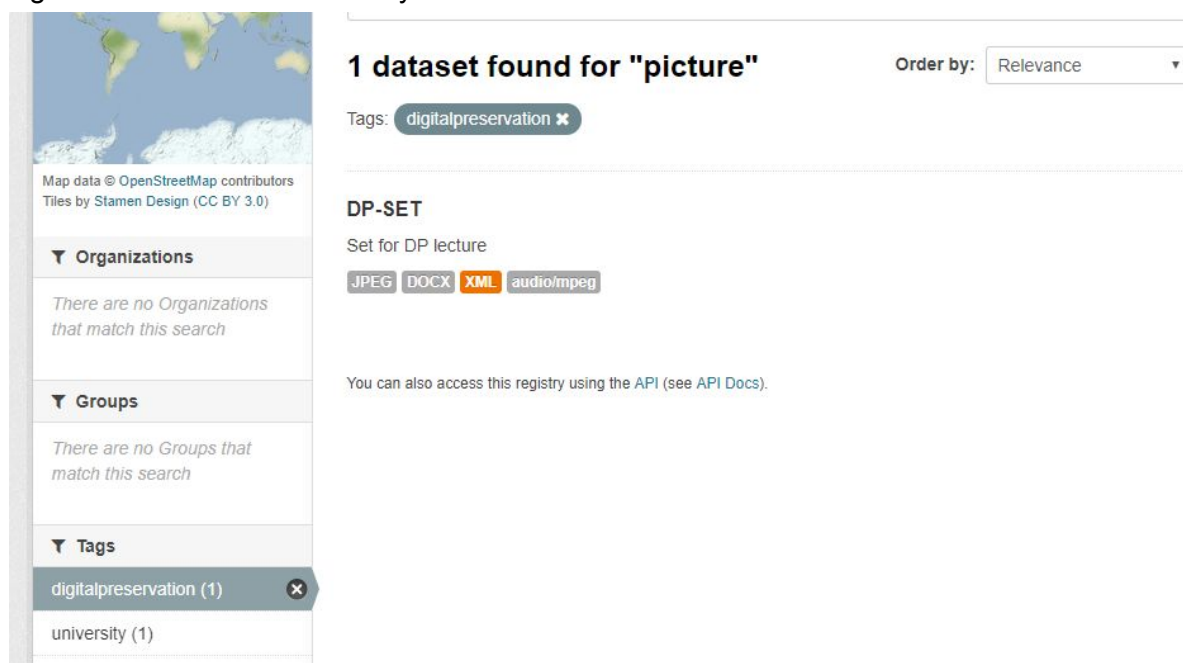
Doe, Joe

Like in CKAN, files like videos, music or microsoft office files can only be downloaded, while pictures or text files can be directly viewed through the UI. There is no preview of single files, only a possible thumbnail for the whole submission, which has to be chosen and uploaded by the publisher.

## Content Search and Discovery

### CKAN

In CKAN you are able to type combinations of words to search for, CKAN then shows you a list of all relevant found Datasets as well as some further possible filters like organization, tags or formats to further filter your search results:



According to our tests, we were able to find our own uploads through all provided metadata (excluding fields like version or last changed time of day).

Faceted search is supported through the filters which can be seen on the left side of the screen, through them one can choose to filter for tags, formats, licenses and more.

The default installation of CKAN does not support full text search within the uploaded documents, but according to the documentation, there are extension that allow the extraction of all file-contents and therefore allow to search through file contents as well.

### DSpace

DSpace allows to search for any combination of words as well, after searching, further filters for specific contents can be set. All results are shown as a list:

## Search



All of DSpace ▼ digpres Go

Hide Advanced Filters

## Filters

Use filters to refine the search results.



Title ▼ Contains ▼

Title  
Author  
Subject  
Date issued  
Has File(s)  
Filename  
File description

1-2 of 2

⊕ ⊖ ⚙

Communities or Collections matching your query

test-digpres  
test-digpres

test  
test digpr

According to our tests, DSpace does not search through all given metadata of the collections. Basically only searching (in the main search bar) for words in the title or description led to the expected result.

Faceted search is supported in DSpace as well, it is called “Discovery” and can be used as a standalone tool to browse through collections as well as in combination with the regular search, as long as the search found more than one relevant collection with different values for each category. Possible filters are author, subject, issue date and more.

Searching for contents of text files is not enabled by default in DSpace but it can be enabled by activating the full text indexing tool of DSpace in combination with the media filters. After the execution of this re-index of DSpace, the content of uploaded files can be searched for.

# User Management and Access Rights

## CKAN

For authentication in CKAN, a user has to enter his username or email-address and his password, at registration an email can be sent to the users mail-address, when this function is activated on the server, by default it is not.

In CKAN you can create new user rules and define the permissions by yourself, but four basic roles for every object are predefined:

- Reader: Can only read objects
- Anon\_Editor: Not logged in user that can edit an object
- Editor: Logged in user that can edit and create objects
- Admin: Administrator

When uploading new data-sets, the uploader can define groups and users that are able to edit this data-set in addition to the uploader. It is also possible to define a data-set as private, which makes it only viewable for the uploader itself.

## DSpace

DSpace uses the mail-address and the password to authenticate the user. As in CKAN an email can be sent to the users mail address while registration but doesn't have to be.

DSpace does not really use a role-based system to give rights to users, except for the site-admin, who has administrator rights.

The user, that creates a new community (group of users), has administration rights to that community. Same goes for the user that creates a new collection of submissions, this user can hand out the submission-rights to the collection for other users.

As in CKAN it is also possible to define a community, a collection or a submission as private, so other users, that don't have submission rights, cannot access it.

### *Addition:*

During our tests we could not find any way to define embargo periods on data, neither in CKAN nor in DSpace.

## Reporting and Administration

### CKAN

CKAN mainly provides every user with his own personal statistics on his profile and dashboard, where things like followers, uploaded datasets or a newsfeed can be found.

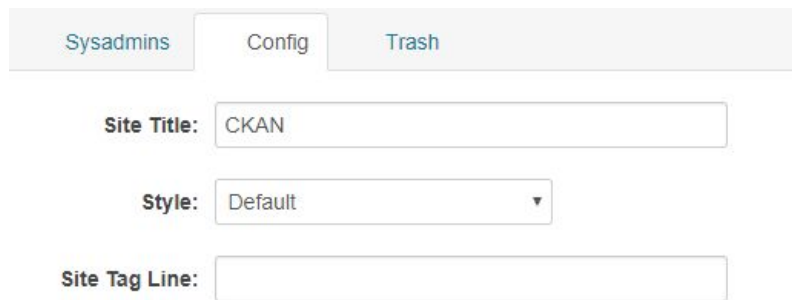
#### **test user**

*You have not provided a biography.*

Followers	Datasets
<b>0</b>	<b>1</b>
Edits	
<b>7</b>	

An administrator account has only access to limited statistics and no central statistics page.

A central administration panel can be accessed through the direct link `<root>/ckan-admin/`, which gives the possibility to change general settings for CKAN as well as the possibility to restore items from the trash.



The screenshot shows a web interface with three tabs: 'Sysadmins', 'Config', and 'Trash'. The 'Config' tab is active. Below the tabs, there are three form fields: 'Site Title' with the value 'CKAN', 'Style' with a dropdown menu showing 'Default', and 'Site Tag Line' which is empty.

This dashboard only gives the possibility to change rather superficial settings, like the site title, style, intro text or a site logo. Further configurations need to be made directly via commands in the console.

## DSpace

In DSpace every administrator account has the possibility to view a “statistics” page, where various different key numbers are shown, i.e. items archived, item views, user logins, most viewed items and many more.

We were not able to find any kind of dashboard or user profile, where every non-administrator could view his own statistics, like it is possible in CKAN.

DSpace also provides a central administration panel for the system, as well as some more administration options, which all can be found in the sidebar when logged in as an administrator.

## Control Panel



The screenshot shows a web interface with five tabs: 'Java Information', 'Configuration', 'SystemWide Alerts', 'Harvesting', and 'Current Activity'. The 'Configuration' tab is active.

Similar to CKAN, the administrative options via the ui only allow limited settings to be changed, more technical changes need to be made directly via console as well.

## Interoperability

### CKAN

#### API

CKAN exposes it's API via a HTTP RPC interface called the [CKAN Action API](#). This API provides access to most of the functionality that is available via the web interface, which includes:

- create, search, update and delete content (items, collections, tags)

- management of organisational entities (users, groups, organisations)

File uploads are managed by a separate api: The [Filestore API](#) provides endpoints to upload file data and hooks to register custom mime types.

### **Metadata**

CKAN does not seem to use any official standard for its metadata. The documentation provides a little bit of information on how to map CKAN metadata to Dublin Core: [Domain Model Dataset](#)

Custom metadata fields can be added by writing a CKAN Extension: [Adding Custom Fields](#)

We've opted for a different and easier approach since we only had to use two custom metadata fields and have just used a dataset's *extra* fields to set the *bitrate* and *duration*.

### **Mime Types / Content Types**

CKAN represents Mime Types as simple text properties of a resource. No special handling is needed to introduce new mime types.

## **DSpace**

### **API**

The DSpace REST API provides a number of endpoints for creating, updating, deleting and querying datasets, collections and related data (collections, communities, metadata schemas, ...).

It does not seem to provide endpoints for user management though.

### **Metadata**

The DSpace API exposes endpoints for querying, creating and updating metadata schemas. This includes predefined schemas like Dublin Core and extends to custom ones like in our case. Metadata schemas and fields have to be defined before they can be set on an item.

### **Mime Types / Content Types**

DSpace requires mime types to be defined up front. Sadly we have not found an API endpoint which allows us to create new mime types. However this was a non-issue in our case since we didn't use any custom mime types. The ones predefined by DSpace were sufficient.

## **Migration Result**

We were not able to successfully migrate all information from CKAN to DSpace.

The DSpace API seemed to for example ignore information like the mime type and file format. Other information like user, dataset id, collection id, etc. were implementation dependent to the respective system and could not easily be transferred.

# Recommendation

Both tools, CKAN and DSpace, are very comparable in what they do, they both provide similar functionality with only minor differences.

The biggest difference, that came to our mind, was setting the systems up on windows.

While both are easy to install on a Unix-based system (Mac OS in our case), CKAN was far more time-consuming to setup on Windows.

## Appendix - Migration Tool

Instructions on how to install and run the tool are available at the tool's Github repository:

<https://github.com/mdietrichstein/digitalpreservation-repo-migration>

## Implementation

Our migration tool is written in Python. It uses the [CKAN Action API](#) to query CKAN repository contents and the [DSpace 6.x REST API](#) to query and create DSpace contents.

## Challenges

The biggest challenge during implementation was the quality of the available documentation, especially regarding the DSpace REST API. It is not always clear which parameters to send as query parameters or form fields or in the json body, etc.

It is for example not documented that passing a name when creating an item has actually no effect on the name that is displayed in the web interface and that the "correct" way to set an item's name is to set its *dc.title* metadata field.

Information like this and other had to be acquired by trial and error and/or searching the web which made the implementation process more cumbersome than necessary.

We were also not able to set a file's mime type or file format. Those fields were accepted by the api, but were not applied to the corresponding uploaded file.

## Workflow Screenshots

The following workflow shows how to use our migration tool to migrate a CKAN repository to a DSpace one.

The CKAN repository is already populated with various files:

- 2 Data Files (CSV and XML)
- 2 Documents (PDF and ePub)
- 2 PNG Images
- 2 Source-Code Files (Java and Go)
- 2 MP3 Audio Files
- 2 MP4 Video Files

The audio and video files are annotated with custom tags (*ti.bitrate* and *ti.duration*). We have used the CKAN *extra* fields to add this information to a repository item.

## List CKAN Repository Contents

```
(digpres) ● migration [master] python migrate.py list-ckan
DigPres Collection
  Data 1
    MimeType: application/xml
    Filename: newyork_babynames.xml
    Description:
  Data 2
    MimeType: text/csv
    Filename: ufo_alcohol.csv
    Description:
  Document 1
    MimeType: application/pdf
    Filename: grimm.pdf
    Description:
  Document 2
    MimeType: None
    Filename: hocus-pocus.epub
    Description:
  Image 2
    MimeType: image/png
    Filename: doggo.png
    Description:
  Image 1
    MimeType: image/png
    Filename: gadse.png
    Description:
  Sourcecode 1
    MimeType: None
    Filename: main.go
    Description:
  Sourcecode 2
    MimeType: text/x-java
    Filename: DataHandler.java
    Description:
  Track 1
    MimeType: audio/mpeg
    Filename: Kosta_T_-_01_---.mp3
    Description:
    ti.bitrate: 320 kbps
    ti.duration: 00:05:26
  Track 2
    MimeType: audio/mpeg
    Filename: Squire_-_01_-_Sense_of_Wonder.mp3
    Description:
    ti.bitrate: 128 kbps
    ti.duration: 00:01:28
  Video 1
    MimeType: video/mp4
    Filename: grow.mp4
    Description:
    ti.bitrate: 320 kbps
    ti.duration: 00:02:37
  Video 2
    MimeType: video/mp4
    Filename: hindenbug.mp4
    Description:
    ti.bitrate: 256 kbps
    ti.duration: 00:01:19
```

## List DSpace Repository Contents

```
(digpres) ● migration [master] python migrate.py list-dspace  
Repository is empty
```

## Run CKAN to DSpace Migration

```
(digpres) ● migration [master] python migrate.py migrate-ckan-to-dspace  
Got dspace session id E8C7741BDB4F07F599BDC9D02B003447  
Registering metadata schema in dspace  
Fetching data from ckan  
Migrating collection "DigPres Collection"  
Community id not specified, using first community: Test (9a253373-e21b-4fd0-b4f7-7d647c691996)  
Migrating package "Data 1"  
Migrating metadata  
Migrating file data  
Migrating package "Data 2"  
Migrating metadata  
Migrating file data  
Migrating package "Document 1"  
Migrating metadata  
Migrating file data  
Migrating package "Document 2"  
Migrating metadata  
Migrating file data  
Migrating package "Image 2"  
Migrating metadata  
Migrating file data  
Migrating package "Image 1"  
Migrating metadata  
Migrating file data  
Migrating package "Sourcecode 1"  
Migrating metadata  
Migrating file data  
Migrating package "Sourcecode 2"  
Migrating metadata  
Migrating file data  
Migrating package "Track 1"  
Migrating metadata  
Migrating file data  
Migrating package "Track 2"  
Migrating metadata  
Migrating file data  
Migrating package "Video 1"  
Migrating metadata  
Migrating file data  
Migrating package "Video 2"  
Migrating metadata  
Migrating file data
```

## List DSpace Repository Contents after Migration

```
(digpres) ➤ migration [master] python migrate.py list-dspace
DigPres Collection
Data 1
  MimeType: application/octet-stream
  Filename: newyork_babynames.xml
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:45Z
  dc.date.available:2018-06-03T12:35:45Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/173
  dc.title:Data 1
Data 2
  MimeType: application/octet-stream
  Filename: ufo_alcohol.csv
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:46Z
  dc.date.available:2018-06-03T12:35:46Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/174
  dc.title:Data 2
Document 1
  MimeType: application/pdf
  Filename: grimm.pdf
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:46Z
  dc.date.available:2018-06-03T12:35:46Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/175
  dc.title:Document 1
Document 2
  MimeType: application/octet-stream
  Filename: hocus-pocus.epub
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:47Z
  dc.date.available:2018-06-03T12:35:47Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/176
  dc.title:Document 2
Image 2
  MimeType: image/png
  Filename: doggo.png
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:47Z
  dc.date.available:2018-06-03T12:35:47Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/177
  dc.title:Image 2
Image 1
  MimeType: image/png
  Filename: gadse.png
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:48Z
  dc.date.available:2018-06-03T12:35:48Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/178
  dc.title:Image 1
Sourcecode 1
  MimeType: application/octet-stream
  Filename: main.go
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:48Z
  dc.date.available:2018-06-03T12:35:48Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/179
  dc.title:Sourcecode 1
Sourcecode 2
  MimeType: text/plain
  Filename: DataHandler.java
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:48Z
```

```
Track 1
  MimeType: application/octet-stream
  Filename: Kosta_T_-_01_-.mp3
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:48Z
  dc.date.available:2018-06-03T12:35:48Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/181
  dc.title:Track 1
  ti.bitrate:320 kbps
  ti.duration:00:05:26

Track 2
  MimeType: application/octet-stream
  Filename: Squire_-_01_-_Sense_of_Wonder.mp3
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:49Z
  dc.date.available:2018-06-03T12:35:49Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/182
  dc.title:Track 2
  ti.bitrate:128 kbps
  ti.duration:00:01:28

Video 1
  MimeType: application/octet-stream
  Filename: grow.mp4
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:49Z
  dc.date.available:2018-06-03T12:35:49Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/183
  dc.title:Video 1
  ti.bitrate:320 kbps
  ti.duration:00:02:37

Video 2
  MimeType: application/octet-stream
  Filename: hindenbug.mp4
  Description:
  dc.contributor.author:
  dc.date.accessioned:2018-06-03T12:35:50Z
  dc.date.available:2018-06-03T12:35:50Z
  dc.identifier.uri:http://localhost:8080/xmlui/handle/123456789/184
  dc.title:Video 2
  ti.bitrate:256 kbps
  ti.duration:00:01:19
```