DCCD system design

**Version: draft**

# Introduction

This document describes the technical aspects of the DCCD (archive) system. Its intended audience is developers that want to get an overview of the system that is not easily captured from the source code.

The Digital Collaboratory for Cultural Dendrochronology (DCCD) is a searchable and international digital archive of tree-ring measurement series, descriptive and interpretative metadata, and associated files such as research reports and photos. Read more about the DCCD at <http://vkc.library.uu.nl/vkc/dendrochronology/research/ProjectsWiki/DCCD%20About.aspx>.

# Architecture

This section describes the system applications, layers, modules, libraries and services and how they interact.

## The web application

The application is implemented as a Java web application and deployed on Tomcat. The user interface layer is build with the Apache Wicket library and depends on the layer with the business logic and data model. That layer uses the TRiDaS libraries for the data model and data conversion (based on the TRiDaS standard). The DANS commons libraries are used for handling the services for archiving (Fedora), user management (LDAP) and searching (Solr).

Layers

* Graphical User interface (GUI)
* Business logic and data model

Libraries

* DANS commons libraries
* TRiDaS libraries

Services

* Fedora Commons
* Open LDAP
* Apache Solr

There are other external applications that DCCD web application should be able to exchange data with. These are TRiCYCLE and TRiDABASE. Both where developed in the same DCCD project and bot exchange data in the TRiDaS format.

The exchange must be done by uploading or downloading from the web application using the corresponding webpages. A more automated programmatic way of data exchange is desirable, but not implemented yet.

## The tools

There are two tools, one to re-index the Sols search index and one general-purpose tool that can be programmed to perform special report, maintenance or repair tasks. Instead of a GUI they have a command line interface, which means you need to run them from a console/terminal. Both tools use the same ‘business logic’ as the web application and thus depend on the web application. Note that the business logic should ideally be placed in a separate library, but that is not the case yet.

# Datamodels and schema

## Archived data

The archival data (files and metadata) is stored using the Fedora Commons repository software (<http://fedora-commons.org/>).

Each DCCD TRiDaS Project is stored in a Fedora ‘digital object’ (sometimes called a dataset or a DataModelObject), which comprises of an identifier, system properties and ‘datastreams’. The content of the archive can be inspected with the Fedora Admin web tool. The datastreams can be separated into the following groups: General Project metadata, TRiDaS entities data/metadata and finally the files.

### General Project metadata

The datastreams with Fedora object metadata. This is partly handled by DANS-commons.

|  |  |
| --- | --- |
| **ID** | **Description** |
| **DC** | Dublin Core   * dc:title TRiDaS Project title * dc:creator TRiDaS Project investigator * dc:subject Titles of all objects of the TRiDaS Project * dc:description TRiDaS Project description * dc:format the fixed string: ‘TRiDaS’ * dc:identifier   The Fedora object id; dccd:[n]   * dc:identifier TRiDaS Project id * dc:language TRiDaS language attribute (iso xxx code) * dc:relation TRiDaS Project references. |
| **PAM** | Project Administrative Metadata   * state * lastStateChange * managerId * legalOwnerOrganisationId |
| **PCM** | Project Creation Metadata   * user   + dccduser     - id     - email     - initials     - surname     - organization     - state     - roles       * role * organization   + dccdorganisation     - id     - state |
| **PPM** | Project Permission Metadata   * defaultlevel * userpermissions  for each item a user id with permissionLevel (JiBX mapping)   + user   + permissionlevel |

### TRiDaS entities

The data being archived is described by the TRiDaS data model, which specifications can be found on the website (<http://www.tridas.org/>). The TRiDaS XML is stored in Fedora, but not as a single file. Instead the data is split into separate entities.

There are eight types of data entities and seven levels of information (both Series belong to the same level). At the highest level we have the ‘Project’ entity and at the lowest level the ‘Values’ (with real measurements, actually data).

Note that the entity streams are stored as fragments of TRiDaS XML.

Entities ordered by level starting with the highest level. Descriptions are obtained from the TRiDaS specification.

|  |  |
| --- | --- |
| **ID** | **Description** |
| **TF[n]** | Project entity Dendrochronological research of a particular object or group of objects |
| **TF[n]** | Object entity The item to be investigated |
| **TF[n]** | Element entity A piece of wood originating from a single tree |
| **TF[n]** | Sample entity A physical specimen or non-physical representation of an element |
| **TF[n]** | Radius entity A line from pith to bark along which the measurements are taken |
| **TF[n]** | MeasurementSeries entity A series of direct, raw measurements along a radius |
| **TF[n]** | DerivedSeries entity Calculated series of values |
| **TF[n]** | Values entity A container for a group of actual measurement values |

The entity tree structure specifies the structure, or parent/child relation of the different entities in XML. In this XML each entity corresponds with an xml element ‘entitytree:<entityname>’ with the ‘title’ attribute and a ‘datastreamId’ attribute containing the datastream ID; TF[n]. The elements are nested to form the hierarchical structure of the TRiDaS data.

|  |  |
| --- | --- |
| **ID** | **Description** |
| **ETS** | Entitytree Contains the hierarchical information of the TRiDaS entities |

### Files

Files, additional and original

Note that a complete TRiDaS file is only stored if it was originally uploaded, otherwise the XML fragments are stored as entities in separate datastreams.

|  |  |
| --- | --- |
| **ID** | **Description** |
| **ASSOCIATED\_FILE\_[n]** | Additional file Mostly these are the reports in pdf, but it could be anything. |
| **ORIGINAL\_FILE\_[n]** | Original file |

Additional files can be uploaded and stored in the repository. The so-called ‘associated files’ need to be specified by using the ‘Files’ element inside the TRiDaS entities for ‘Project’ to ‘Sample’ level.

Files with measurements in a non-TRiDaS format can be uploaded and will be converted and added to the stored TRiDaS. For this the TRiDaS ‘genericField’ element is being used. This element should be placed inside a ‘measurementSeries’ XML element and specifies the measurement values file to use. Example for a file with the name "mydata ":

<genericField name="dccd.treeringdatafile">mydata</genericField>

The upload mechanism will detect this and ask for the files it refers to.

To indicate that the data is converted and added to the TRiDaS data in the repository the XML in the repository is changed to:

<genericField name="dccd.treeringdatafile.uploaded ">mydata</genericField>

When the TRiDaS XML is downloaded it then does not specifies that the files need to be added, but instead that they already have been added. The ‘original’ TRiDaS file is of course stored unchanged.

## User data

This is the data stored in the LDAP database (Open LDAP). Use the Apache DS Studio client, for instance to inspect it.   
DCCD has users and organizations. The organizations are used for grouping users; more than one user can be member of an organization. In DCCD users should be member of only one organization. Organizations also play a role when ownership of a project is changed. Ownership can only be transferred to another member of the same organization thereby always keeping the project under the initial organization.

**LDAP Structure**

* dc=dans,dc=knaw,dc=nl
  + ou=dccd
    - ou=organisations
    - ou=users

Organisation (objectClass = dccdUserOrganisation)

|  |  |
| --- | --- |
| **Name** | **Description** |
| **ou** | the id |
| **dansState** | See: user dansState |
| **uniqueMember’s** | The uid’s of the members |

User (objectClass = dansUser, dccdUser, inetOrgPerson etc.)

|  |  |
| --- | --- |
| **Name** | **Description** |
| **uid** | the user id (login name) |
| **cn** | common name = surname, intials |
| **sn** | surname |
| **o** | organisation id |
| **mail** |  |
| **initials** |  |
| **title** |  |
| **givenName** | firstname |
| **ou** | department |
| **employeeType** | function |
| **postalAddress** | address |
| **postalCode** |  |
| **l** | city |
| **telephoneNumber** | telephone |
| **displayName** |  |
| **userPassword** | SHA hashed, original password string is not stored and thus not visible to anyone even admin |

DANS specific, but also used by DCCD

|  |  |
| --- | --- |
| **Name** | **Description** |
| **dansAltTel** | alternativeTelephone |
| **dansAcceptConditionsOfUse** | boolean |
| **dansLastLogin** | date |
| **dansnewsletter** | Boolean  Unused by DCCD! |
| **dansState** | REGISTERED, CONFIRMED\_REGISTRATION?, ACTIVE, BLOCKED DCCD uses different labels in the GUI! |

DCCD specific

|  |  |
| --- | --- |
| **Name** | **Description** |
| **dccdDAI** | Digital Author Identifier, but just handled as a string |
| **dccdRoles** | USER, ARCHIVIST, ADMIN but ARCHIVIST not used!, and always USER. |

For the DCCD web application to work, at least one user with the ADMIN role should be instantiated in the database. For this the id ‘**dccduseradmin**’ is reserved and this account’s email address is also used for mailing the membership registration request.

## Project states

The TRiDaS ‘Project’ (and all it contains) is stored in a ‘digital object’ and that is in a certain state. This state and the permission settings described in the next section determine what and who can view and edit the content.

The states are being used are shown below.

|  |  |
| --- | --- |
| Name | Description |
| DRAFT | Draft  initial state after ingestion into the archive; upload  NOTE. In this state the data can be ‘deleted’!  Also Note that it can be edited, but not the structure! |
| PUBLISHED | Archived From Draft and only when successfully validated!  NOTE: explain extra restrictions!  Can’t be edited or deleted. |

Archiving (or publishing) is a very important step, which involves validation of the data to make sure that it is fit to be published.

The data must conform to the TRiDaS schema (version 2.2) plus some extra restrictions that improve consistency of the archive. The project identifier (combined with the domain attribute) should be unique and all the ‘type’ fields should be using the terms from the controlled vocabulary described at the end of this document.

## Permission settings

The Project’s permission settings determine which information is visible to the users. The Project’s status is also taken into account; when in Draft it is more restricted. Besides the project owner who is working on the ‘draft’ only a user with the ADMIN role is allowed to see it.

|  |  |
| --- | --- |
| Name | Description |
| minimal | Maximum security only some of the project level information is allowed.  Note that entity titles are also always visible to logged-in members! |
| project | All Project information allowed |
| object | All project and object info allowed |
| element | Project to and including element |
| sample | Project to and including sample |
| radius | Project to and including radius |
| series | Project to and including measurement series and derived series |
| values | Complete TRiDaS allowed, also download possible |

When a Project is ingested (uploaded) it should be secure without any interaction, therefore the initial default permission is ‘minimal’.

Default permission level: **minimal**

Maximum security (**minimal** permission level): public metadata is shown and can be searched. All other content is inaccessible.

The public metadata or ‘open access’ data consist of:

* Project title
* Project identifier
* Project type
* Principal investigator
* Period of research
* Type of material that was studied (archaeology, ship's archaeology, furniture et cetera)
* Laboratory
* Object title

Except for the Object title all public metadata comes from the Project level.

The **project** to **series** permission level: in addition most of the content of the selected TRiDaS level is shown and can be searched.

The **values** permission level: this level permits downloading of complete project files, including measurement series and associated files, in TRiDaS XML and other dendrochronological data formats.

Note that permission settings can be changed after archiving.

Although the aim is to have owners relax the permissions on their data they could make them more restrictive over time.

Also note that ownership can be changed (see section?).

## User states

What users can see or do depends on the user state. All ‘open access’ information is visible also to visitors of the website that are not logged in and thus possibly non-members. Only users with the ACTIVE state can see more than the ‘open access’ information. For normal (non ADMIN) users there can be further restrictions specified for each published Project by the owner or ADMIN by setting Project permission settings (see section?).

Visibility for Published data

|  |  |  |  |
| --- | --- | --- | --- |
|  | **All** | **Use permission** | **Only ‘open access’** |
| **ADMIN role (active user)** | x |  |  |
| **ACTIVE user (non-admin)** |  | x |  |
| **Not ACTIVE or non member** |  |  | x |

For unpublished data (DRAFT state) the rules are different. ACTIVE users can upload data and can see and edit their own projects. Users with the ADMIN Role can see (and edit?) everything, also data from other users in DRAFT state.

When a user is being registered or activated the organization will also be registered or activated if wasn’t already. Organizations cannot be blocked, because it can have active members.

dansState

|  |  |
| --- | --- |
| Name | Description |
| REGISTERED | Cannot login After registration, users are in this initial state. |
| ACTIVE | Can login This must be set by a user with the ‘Admin’ role.  Note that when BLOCKED the account can be activated again. |
| BLOCKED | Cannot login  This must be set by a user with the ‘Admin’ role.  The user data is never deleted, because he or she may own ‘projects’ stored in the archive while the account was active.  It can be set to active again. |

dansRole

|  |  |
| --- | --- |
| Name | Description |
| USER | Regular User Can login and edit own settings and Projects. |
| ADMIN | Administrator Can also edit settings and Projects of other users and manage users state Active/Blocked. Note the ‘dccduseradmin’ account. |

## Search index

This section describes what fields are being indexed by the search service to enable the web application to search for ‘Objects’ or ‘Projects when in ‘My Projects’.

Permissions play an important role in searching, because you can infer knowledge by searching and inspecting the results or even the absence of results. What a user can find therefore depends on the permission, status and role; see ‘user states’ section above.

The indexed fields that are used to facilitate a filtered search that satisfy the permissions are: type, ownerId, administrativeState and permissionDefaultLevel.

Note that for searching the defaultPermission level is being used and not the user specific permission levels. This was to complicate to implement for searching, but since the defaultPermission is meant to be less restrictive inferring knowledge that should not be ‘visible’ should not be an issue.

Fields used for retrieving Projects and Objects from Fedora Repository.

|  |  |
| --- | --- |
| Name | Description |
| ID | Unique key for index.  Note that we cannot use the Project.identifier as key because in Draft state it can still change.  ???  **TODO** find out |
| PID | A persistent, unique identifier for the Fedora object.(but not our resolvable persid with urn etc. ???)  **TODO** find out |
| DATASTREAMID | Used for retrieving the Object from Fedora Repository. But only if it is an object; see ‘type’ field. |

Other non-TRiDaS fields mainly used for permission and administration purposes.

|  |  |
| --- | --- |
| Name | Description |
| type | Specifies if it is a Project or Object For an Object all its project information is also indexed.  The value is either dccdProject or dccdObject. |
| lat | Latitude of the first Object of the Project. WGS84 coordinate. |
| lng | Longitude, see above |
| ownerId |  |
|  | See dataset state |
| administrativeStateLastChange | Can be useful for sorting results. |
| permissionDefaultLevel | See Permissions |
| tridas.project.identifier .exact | Need to match and get the exact string from a search result in order to determine uniqueness. |
| tridas.project.identifier .domain.exact | See above. |

Fields by TRiDaS entity level

tridas.project

|  |  |
| --- | --- |
| Name | Description |
| title | Open access |
| identifier | Open access |
| comments |  |
| type | Open access |
| type.normal | Open access |
| description |  |
| laboratory.name | Open access |
| laboratory.name.acronym | Open access |
| laboratory.address.cityOrTown | Open access |
| laboratory.address.country | Open access |
| category | Open access |
| category.normal | Open access |
| category.normalTridas | Open access |
| investigator | Open access |
| period | Open access |
| research.identifier |  |
| research.identifier.domain |  |
| research.description |  |

tridas.object

|  |  |
| --- | --- |
| Name | Description |
| title | Open access |
| identifier |  |
| comments |  |
| type |  |
| type.normal |  |
| description |  |
| creator |  |
| coverage.coverageTemporalFoundation |  |
| location.locationType |  |
| location.locationComment |  |
| genericField |  |

tridas.element.

|  |  |
| --- | --- |
| Name | Description |
| title |  |
| identifier |  |
| type |  |
| type.normal |  |
| description |  |
| taxon |  |
| shape |  |
| altitude |  |
| slope.angle |  |
| slope.azimuth |  |
| soil.description |  |
| soil.depth |  |
| bedrock.description |  |

tridas.sample

|  |  |
| --- | --- |
| Name | Description |
| title |  |
| identifier |  |
| type |  |
| type.normal |  |
| samplingDate |  |

tridas.radius

|  |  |
| --- | --- |
| Name | Description |
| title |  |
| identifier |  |
| woodCompleteness.pith |  |
| woodCompleteness.heartwood |  |
| woodCompleteness.heartwood .missingHeartwoodRingsToPith |  |
| woodCompleteness.heartwood .missingHeartwoodRingsToPithFoundation |  |
| woodCompleteness.sapwood |  |
| woodCompleteness.sapwood .nrOfSapwoodRings |  |
| woodCompleteness.sapwood .missingSapwoodRingsToBark |  |
| woodCompleteness.sapwood .missingSapwoodRingsToBarkFoundation |  |
| woodCompleteness.sapwood .lastRingUnderBark |  |
| woodCompleteness.bark |  |

tridas.measurementSeries

|  |  |
| --- | --- |
| Name | Description |
| title |  |
| identifier |  |
| analyst |  |
| dendrochronologist |  |
| measuringDate |  |
| measuringMethod |  |
| measuringMethod.normal |  |
| interpretation.provenance |  |
| interpretation.deathYear |  |
| interpretation.firstYear |  |
| interpretation.lastYear |  |
| interpretation.pithYear |  |
| interpretation.statFoundation .statValue |  |
| interpretation.statFoundation .usedSoftware |  |
| interpretation.statFoundation .type |  |
| .interpretation.statFoundation .type.normal |  |
| interpretation.statFoundation .significanceLevel |  |
| interpretationUnsolved |  |

## Controlled vocabularies

Controlled vocabularies are used for multi lingual searching and also important when validating for archiving. The vocabulary terms files are stored as UTF8 tab delimited text files and bundled with the web application.

For each ‘type’ of terms there is a table. The columns correspond to the different languages; each column contains the terms of one language. Each row (or record) contains the terms that are to be considered equivalent. The same term can be in several rows for the same language because it ‘maps’ to different terms in another language. This is in particular the case with the elaborate table for Object and element types.

List of term files

|  |  |
| --- | --- |
| Name | Description |
| object.type.txt | Elaborate table of equivalent terms |
| project.category.txt | Small table |
| project.type.txt | Small table |
| sample.type.txt | Small table |
| derivedseries.type.txt | Small table |
| measurementseries.usedsoftware.txt | Small table, only English |
| element.taxon.txt | Taxonomy (one language but with hierarchy) |

Multi-lingual searching is implemented by using the records of vocabulary terms as synonyms (equivalent terms). This is not perfect because you may find more then you want when he same term has a different meaning in different languages but in practice it works really well.

The Taxon terms (element.taxon.txt) are organized differently. Instead of being multi-lingual the terms are in Latin. Furthermore the hierarchy is important and also being used in the GUI of the web application. It could be used in query expansion by adding the narrower terms to a given query.

The first column contains the ‘node’ number the second column the ‘node’ name and the third and last column the parent node number.