1

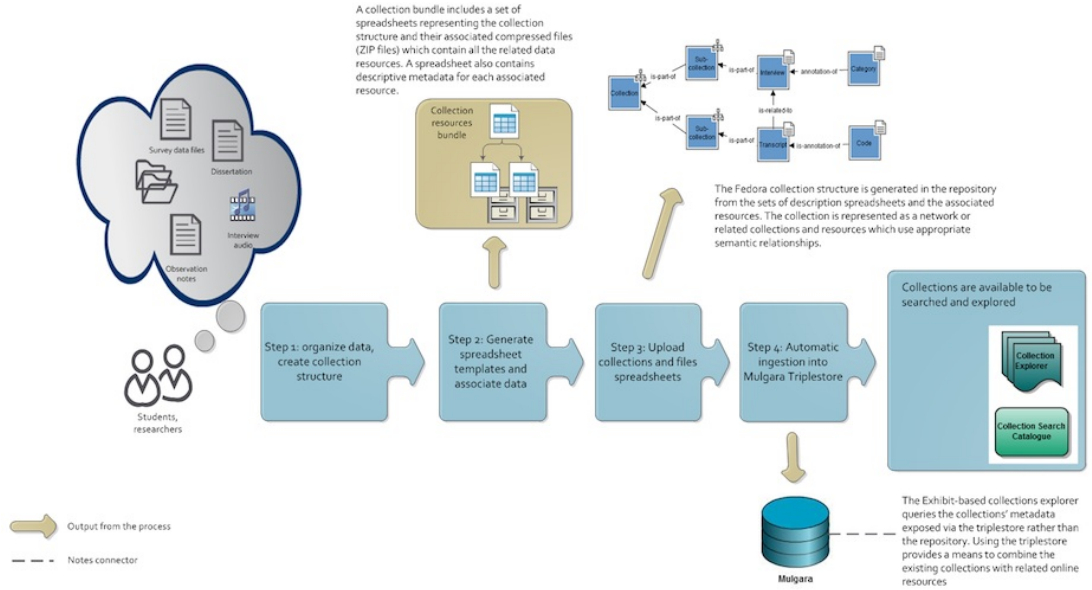
TheUsers

Researchers: QuDEx Repository Toolkit user guides

Updated Yesterday (21 hours ago) by amg...@gmail.com

Archiving Qualitative Collections

The developed toolkit provides mechanisms to create research collections in simple ways and by following processes with which students and other researchers who are not archivists or database designers are familiar. The figure below shows the process of organising, documenting and creating an archive with the support of the toolkit. A 'collection' can be seen as a set of resources associated with a project, dissertation or article, data are primarily collected materials (survey data, interview audio files, transcripts or notes) and analytical outputs may be numerical summaries, generated charts or tables, and completed pieces of writing. These are collated using spreadsheets either in Excel or CSV formats, and once stored into the archive, they are organised in a manner that is very similar to a 'files and folders' structure on a desktop computer.



Caption: framework for data organisation, collection generation and publishing

The toolkit guides the users (students, researchers) through the process of structuring and describing their data using a series of web applications (steps 1, 2 and 3 in the diagram shown above) without their having to deal directly with the internal working of the underlying systems (Fedora or Mulgara), or to understand too much about the metadata schemas used in the toolkit. The first steps in the collection creation process are concerned with describing and organising the data (step 1 in the diagram above) in appropriate ways, to ensure that the data can be shared and integrated with existing online content. As it has been previously highlighted, standardised data documentation is key to digital archiving and has an impact on dissemination, data discovery and effective re-use. To ensure that a research study and all its associated materials are described in consistent ways, the first stage of the archiving process, data organisation and documentation, is supported by the template builder application. This application is a spreadsheet-based tool for capturing metadata about qualitative and mixed-methods studies, providing an easy-to-use mechanism for describing all the materials included in a particular project or collection (step 2 in the diagram above).

Once the descriptive spreadsheets have been created and populated with descriptions of the data, research instruments, analytical tools and outcomes of interpretation of analysis, the user can then generate and archive the collection in the repository (step 3 in the diagram above). This involves using another web application, 'QuDEx collection manager', which allow users to, firstly, generate the collection structure and secondly, upload all the data resources with which the collection is associated. The process of uploading the spreadsheets in the repository transforms each described element into the data structures included in the model. The 'collection structure' spreadsheets are used to generate collection and sub-collection elements representing the organisation of the research collection and the 'resources' spreadsheets are used to generate files objects representing the different instruments produced during the research process.

User guides

- Using the Template Builder: document your research data
- Using the Collections Manager: store your research collection in the repository
- Using the Collections Explorer: visualise your research collection

SpreadsheetBuilder

QuDEx Repository Tools: Spreadsheet builder Walkthrough

Updated May 30, 2013 by amg...@gmail.com

QuDEx Template Builder

This application is a web application underpinned by a set of java libraries to assist users with the organisation and documentation of their data. It presents users with a stage-by-stage process, beginning with their selection of metadata terms that are necessary to describe their data. This process generates a set of spreadsheets that can then be completed to express both resources and collection structures while avoiding the tendency for individuals to improvise metadata fields or to use terms from established standards in inconsistent ways.

Template spreadsheets

A research collection can be archived in the repository from two different types of spreadsheets: Collection spreadsheet and Files spreadsheet.

The first of these, Collections/sub-collections spreadsheet, is used to populate a research study collection structure in an instance of Fedora: each row of a collections spreadsheet represents one sub-collection within a hierarchical structure. These collection objects will hold their own metadata description fields and a special datastream, which holds the QuDEx instance generated to describe the whole archived collection. This XML instance is contructed from all the file objects that belong to a specific collection in the repository. These objects are stored in the repository as fedora digital objects, and then they have been mapped to their associated XML fragment under the QuDEx schema.

The second type of spreadsheet, Files spreadsheet, contains descriptions of "QuDEx Files" that will be stored in Fedora as objects. Each of these files will be associated with an already existing collection/Sub-collection object and will hold the following data:

- File specific metadata fields to describe this document
- File QuDEx Categories
- File QuDEx Codes
- File QuDEx Categories

Creating a spreadsheet involves the following steps:

- Select the type of spreadsheet: Collections/Sub-collections or Files spreadsheet.
- Metadata configuration: Adding descriptive headers to the spreadsheet
- Review and Generate spreadsheet

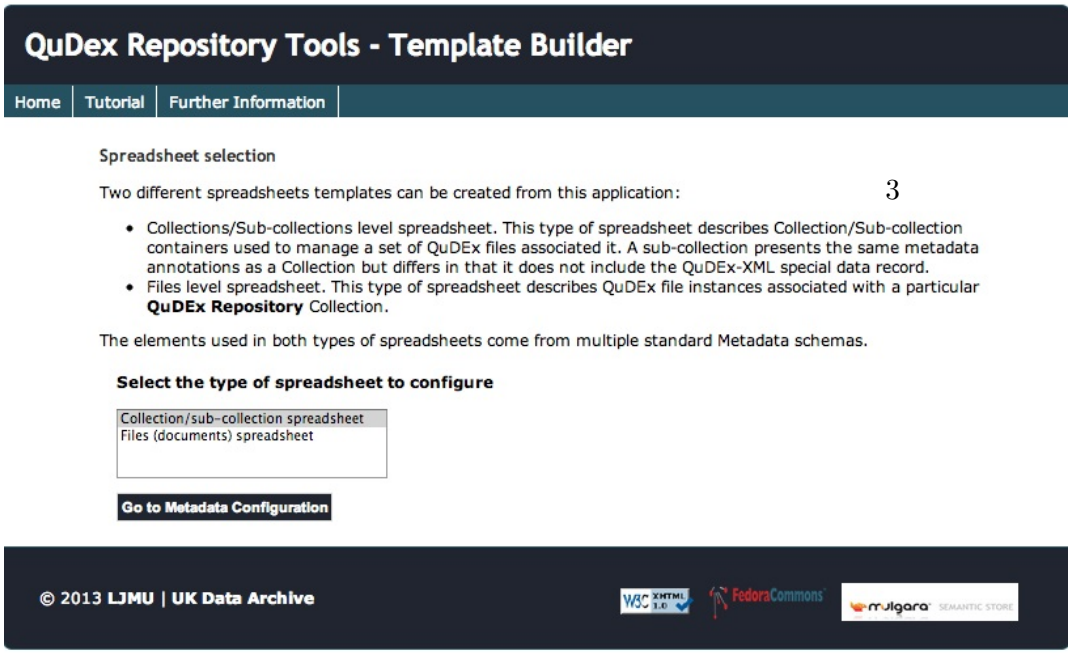
Application Walk-through

In this example we are creating a QuDEx file-level spreadsheet, which uses the following Metadata Schemas:

- RDF-ised QuDEx Schema. This custom vocabulary include those QuDEx specific fields that do not have their equivalent in other standard schemas
- Dublin Core. Vocabulary used for those QuDEx fields that can be mapped to Dublin Core elements
- W3C WGS84 lat/long Vocabulary. Vocabulary used to describe geographical locations by place, or latitude/longitude pairs.
- RDF Vocabulary. To specify the type of the different types of objects included in a collection, e.g. collection, file.
- Fedora Relationships Ontology. To specify the different relationships between the resources within a collection.

Step 1 - Select the type of spreadsheet

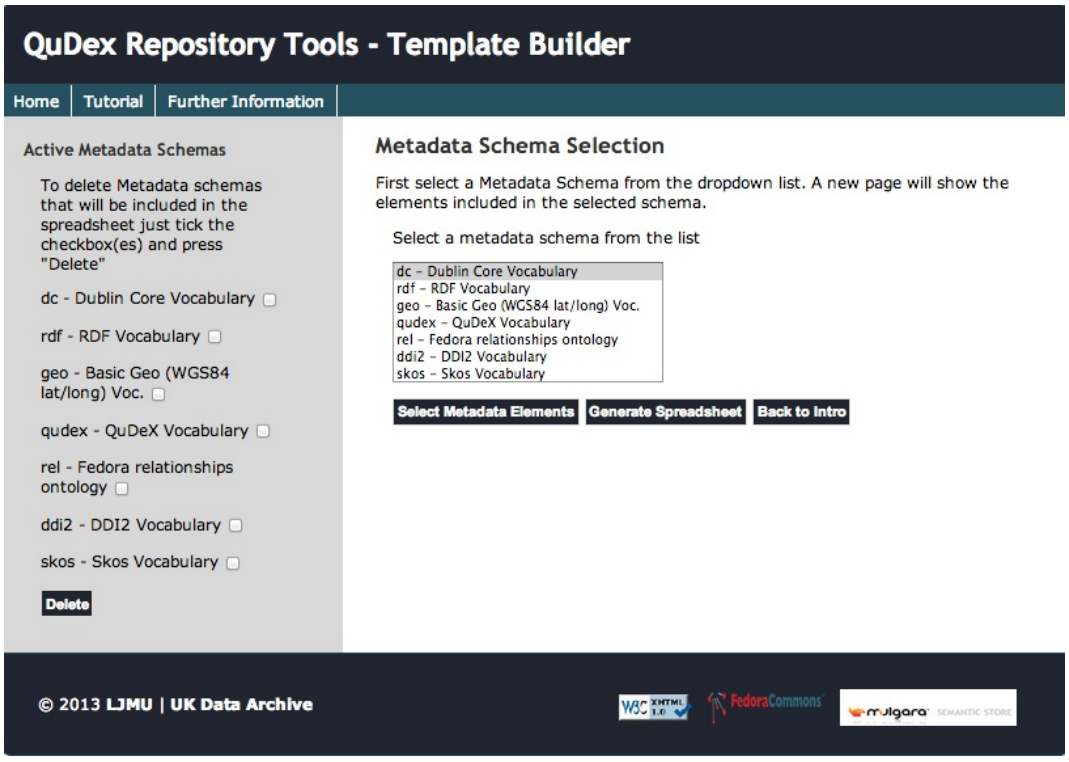
Once we have started the process, the first step is selecting the type of spreadsheet for which we will generate the template. For this example the selection is "Files spreadsheet" and then we click on 'Go to Metadata Configuration'



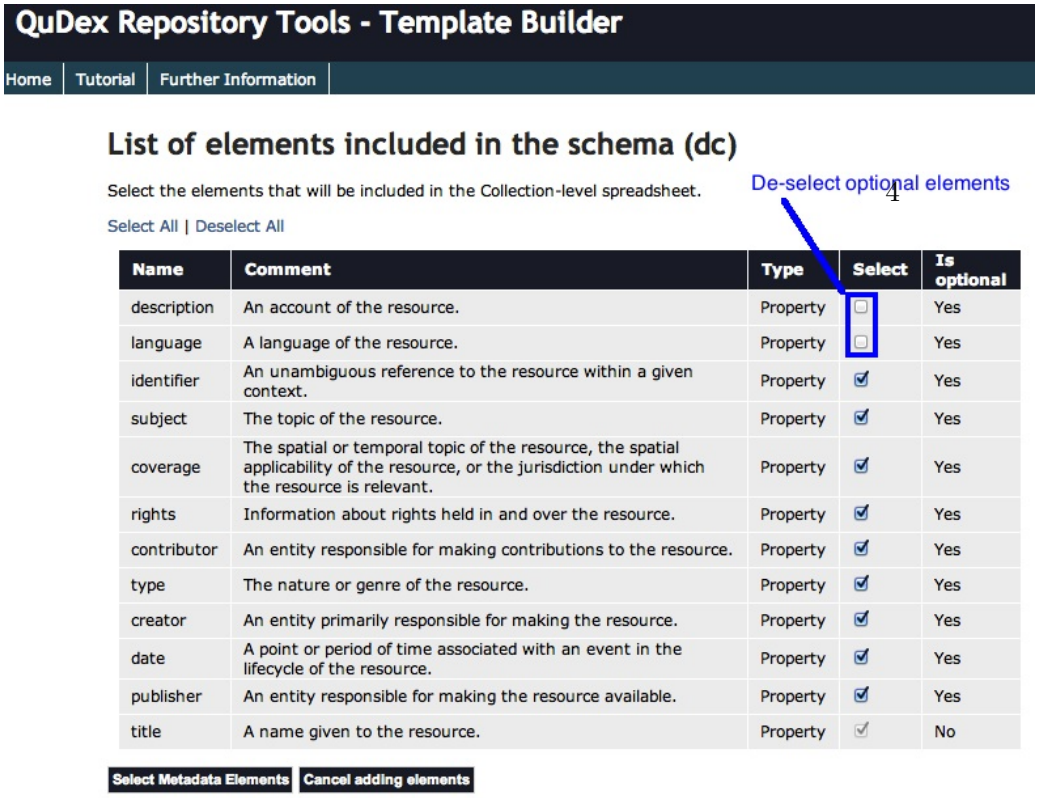
Step 2 - Metadata Selection

The next step is selecting the different metadata elements (spreadsheet headers) that we will be using to describe the different resources within the collection (**Metadata Selection** screen). In the sidebar on the left we can see a list including the different metadata schemas that have been pre-selected. To ease the process, all the available elements (optional and mandatory) have been pre-selected initially. Additionally from here, specific schemas can be deleted from the spreadsheet: e.g. we won't be using relationships therefore no elements from the Fedora ontology are needed. If the schema we want to delete includes elements that are compulsory, only those will remain in the spreadsheet.

Note: If we want to include all the fields, then we can jump directoy to the review section.



To edit what elements from a specific schema will be included in the template spreadsheet we can select the schema we are interested in and then we will be redirected to a page displaying a table which included the different elements included in the selected schema, a brief description of the elements and whether they are optional or compulsory.



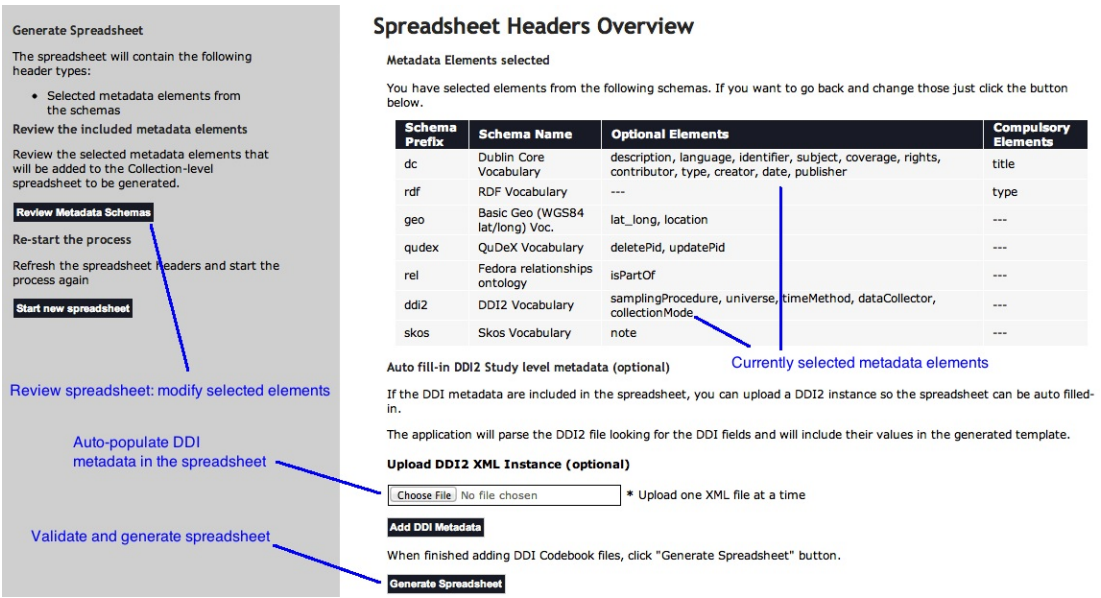
Step 3 - Review and Generate spreadsheet

Different actions could be performed here depending on what type of spreadsheet template we are generating.

If we were generating a 'Files spreadsheet' template then the actions that could be performed are the following:

- Review the metadata headers included in the spreadsheet so far. At this point, you could always go back and delete or add new elements.
- Uploading resources ZIP files. For files spreadsheets, some information about the resources that will be described in the spreadsheet can be auto-populated into the spreadsheet. Examples of such information include file name, mimetype, creation data or even geo-location information.

For a 'Collections spreadsheet', the DDI information at collection level can be auto-populated into the generated spreadsheet. This can be performed by uploading an Atlas.ti XML instances that contains DDI information about the study - e.g. Data collection mode, sampling strategies, data collector(s), etc.



The last step, once reviewed the metadata headers that will be included, is to generate the spreadsheet. Once we click on the 'Generate spreadsheet' link. The spreadsheet is created, validated and downloaded automatically.

For a simplified description of the process of creating a template see: Activity Diagram

DataDocumentation

Spreadsheet documentation

Updated Jun 4, 2013 by amg...@gmail.com

Introduction

A key part of making data user-friendly, shareable and with long-lasting usability is to ensure they can be understood and interpreted by any user. This requires clear data description, annotation, contextual information and documentation.

Data documentation explains how data were created, their meaning and what their content and structure are. It ensures that data can be understood during research projects, that researchers continue to understand data in the longer term and that re-users of data are able to interpret the data. Good documentation is also very important for data preservation purposes.

The process of documenting the data in appropriate ways can be simplified if performed at the beginning of a project and continued throughout the research. It also considered as part of best practice in creating, organising and managing data.

Good documentation for research data contains both study-level information about the research and data creation, as well as descriptions and annotations at data item or data file level.

Metadata are a subset of core data documentation, which provides standardised structured information explaining the purpose, origin, time references, geographic location, creator, access conditions and terms of use of a data collection. Metadata are typically used for resource discovery, providing searchable information that helps users to find existing data, as a bibliographic record for citation, or for online data browsing.

Collection spreadsheet

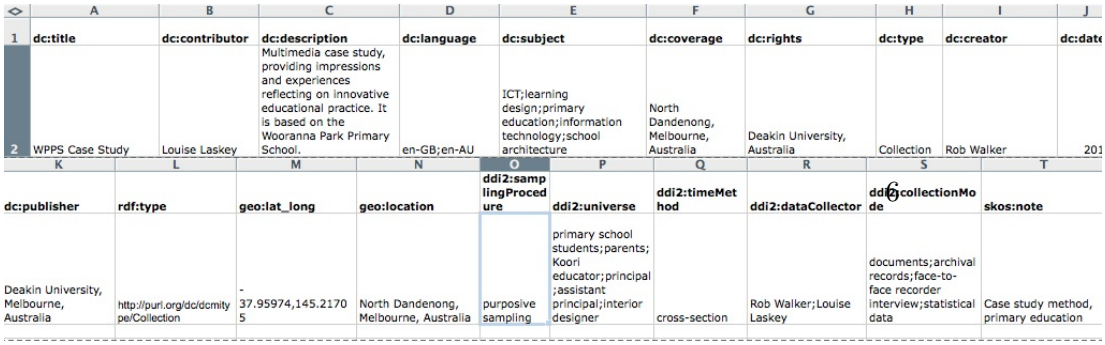
When creating collection spreadsheets, two different types of information are being provided. The first one gives information about the collection structure, that is how the collection is organised. The second type of information is related to the research study as a whole (study-level documentation) and provides an overview of the research context and design; data collection methods; data preparation and results or findings.

The information about the structure of the collection is provided in the spreadsheet itself, i.e. the collection structure is normally hierarchical - the rationale for this is that most people tend to organise their data in the form of hierarchical folders structure - and all the collections included in a specific level of the hierarchy are described and included in the same spreadsheets.

The study-level information is provided in each collection spreadsheet by using metadata terms from a set of standard vocabularies or metadata schemas. The information related with the context of the study can be specified by using a combination of elements from DDI and Dublin Core schemas. For simplicity, only a small set of terms from DDI has been selected:

Element	DDI comment	Description
samplingProcedure	The type of sample and sample design used to select the survey respondents to represent the population	The DDI description covers mostly quantitative data, however, qualitative sampling procedures such as purposive sampling, or representative sampling covering time, place and people. When using this field a short description should be use.
universe	The group of persons or other elements that are the object of research and to which any analytic results refer	The participants of the research, e.g. final year undergraduate students; primary school teachers...
timeMethod	The time method or time dimension of the data collection	This field is more suitable for quantitative studies, where the participants are followed over periods of time, e.g. longitudinal studies. However, for qualitative studies, using interviews for example, a short description of the data collection could be included
dataCollector	The entity (individual, agency, or institution) responsible for administering the questionnaire or interview or compiling the data. This refers to the entity collecting the data, not to the entity producing the documentation	The person performing the data collection: researcher, research assistant, etc.
collectionMode	The method used to collect the data; instrumentation characteristics	This is probably one of the most important fields and a short description of the different data collection methods should be included here. For more detailed information, one can attach the research design as one of the documentation materials included in the collection. Examples of this field include semi-structured interviews, participant observation, online survey, etc.

For additional study-level documentation, some of the DDI terms can be mapped to Dublin Core elements. For more information, the DDI provides a recommended mapping.



	A	B	C	D	E	F	G	H	I	J
1	dc:title	dc:contributor	dc:description	dc:language	dc:subject	dc:coverage	dc:rights	dc:type	dc:creator	dc:date
2	WPPS Case Study	Louise Laskey	Multimedia case study, providing impressions and experiences reflecting on innovative educational practice. It is based on the Wooranna Park Primary School.	en-GB;en-AU	ICT;learning design;primary education;information technology;school architecture	North Dandenong, Melbourne, Australia	Deakin University, Australia	Collection	Rob Walker	201

	K	L	M	N	O	P	Q	R	S	T
1	dc:publisher	rdf:type	geo:lat_long	geo:location	ddi2:samplingProcedure	ddi2:universe	ddi2:timeMethod	ddi2:dataCollector	ddi2:collectionMode	skos:note
2	Deakin University, Melbourne, Australia	http://purl.org/dc/dcmitype/Collection	-37.95974,145.21705	North Dandenong, Melbourne, Australia	purposive sampling	primary school students;parents;Koori educator;principal;assistant principal;interior designer	cross-section	Rob Walker;Louise Laskey	documents;archival records;face-to-face recorder interview;statistical data	Case study method, primary education

Caption. Example of a collection spreadsheet with study-level information

Additionally some QuDEx elements are used to support the application's collection updating capabilities. The only documentation required to perform such operation are the assigned Fedora PID (Physical Identifier) that when the collection (and associated sub-collections) were stored in the repository. The QuDEx fields required to perform the update/delete operations are: 'updatePID' and 'deletePID'. The first is used to locate within the repository the collection (sub-collection) that the user wants to update. By updating here, we mean only modifying the collection's metadata. No update operations at file or resource level are supported in the system. For more information regarding the latter see the update collection wikipage. The second is used to locate within the repository the collection to be deleted. All the sub-collections and files associate with this collection will be also removed from the repository. Examples of valid Fedora PIDs include: 'qudex:00101', where qudex is the namespace followed by a number.

The last type of information at collection-level is related to location, to provide richer contextual information. Two different elements can be used to express geographical information (both from the GEO lat/long vocabulary). The first of these is 'location' which represent the location by using strings such as United Kingdom, or North Wales. The second can be more specific and used to populate visualisation tools using maps. It is specifying the location by using a pair of latitude longitude coordinates.

Files spreadsheet

File spreadsheets contain data-level documentation for the resources included in a collection. Three different types of information can be provided. The first type gives information about the data resources themselves, that is information such as title for the file, short description, keywords, creator - mostly information that can be described using Dublin Core terms - and physical information about the file that can be extracted automatically. Indeed, when uploading resource compressed files into the template builder application, the generated templates auto-populate such kinds of information. Examples of these include: file mime type, creation date, file size, geographical location (for images that use EXIF metadata), etc.

The second type of information is analytical information that applies to individual files within the collection. This kind of information is documented by using elements from the QuDEx schema. The following table summarises these elements and provides examples of use. The original use of the analytical information within the QuDEx schema is for supporting the analytical information managed in CAQDAS packages.

Element	Description	Example
labelCode	The label of a qudex Code element. For this field user-friendly or comprehensive words associated with the codes should be used.	In the following example, the values for labelCode should be the short sentences or words associated with a numerical code. See example 1.
code	The value of a qudex Code element. For this field the actual numerical code value should be used.	In the following example, the values for labelCode should be the short sentences or words associated with a numerical code. See example 1.
labelCategory	The label of a qudex Category element (source QuDEx schema). The same as with labelCode.	See category example 2
category	The value of a qudex Category element. The same as with codes.	
labelMemo	The label of a qudex Memo element.	See category example 2
memo		
isOriginal	Custom element not included originally in QuDEx that is used to identify whether the resource will be stored in the repository or a reference to the original file will be stored instead, e.g. a URL that points to the location of the original file	isOriginal=true means that the spreadsheet will contain a header called 'originalReference:R' which is interpreted as a link to an external resource that lives outside the repository; in contrast, isOriginal=false generates a header 'sourceReference:M'. M and R qualifier are Fedora specific and refer to the type of datastream that will hold the resource being uploaded. M for managed (internal to the repository) and R for redirected (a reference to the original location is provided.)

Coding scheme and QuDEx elements example

In this example the coding scheme is composed of a hierarchy of codes, expressed numerically. The numerical codes are the values that are documented with the 'code' element. Each numerical code has a short description attached to it, providing meaning to the code. The short description is normally either a short sentence (like in the example) or a word. The description of the code is then documented by using the 'labelCode' term.

1. The smoking period
    1. When started and stopped
    2. Reasons for starting smoking and continuing to smoke
2. Influences on smoking behaviour
    1. Home environment
    2. Peers
    3. Work environment

(source for the example: http://www.uniteforsight.org/global-health-university/quantify-research)

labelCode = Reasons for starting smoking and continuing to smoke

code = 1.2

Categories and QuDEx elements example

The mechanics for expressing categories is very similar to the ones for codes. The main difference lies on the explicit management of hierarchies

for categories. For codes, there is no explicit way of expressing the hierarchy (when working with a hierarchical coding scheme), therefore they won't be displayed as a hierarchical tree when visualising them in the Collection Viewer. In contrast, for categories there is a mechanism to express the hierarchy in the template spreadsheet. The way of expressing the hierarchy is as follows:

Let's use as an example for categories 'Sex' and sub-categories 'Male' and 'Female'.

- Sex
  - Male
  - Female

Each element within the category hierarchy or tree has to be represented by its full path. The different levels of the hierarchy are specified by using the character **':' (colon)**. The values for the categories are **S** for sex; **M** for male and **F** for female; and the labels (labelCategory) are sex, male and female respectively. Each element being specified is separated by **';' (semicolon)**. Then, in the spreadsheet, the labelCategory for the example above would be

labelCategory = sex;sex:male;sex:female

category = S;S:M;S:F

and the matchings are sex to S; sex:male to S:M and sex:female to S:F.

Note It is extremely important that both labelCategory and category fields contain the elements in the same order otherwise they won't be associated correctly when generating the categories in the repository.

Relationship information

The third type of information is relationship information that is used to link resources and express the semantics of their relationship. This type of information is both when visualising the collection elements and when sharing data. Useful uses of these relationships include linking transcriptions with their associated audio/video file; linking documentation such as notes with specific analysis documents, etc. The following table summarises the types of relationships that can be included in the collection. While the other types of documentation - analytical and data level documentation - are included via the spreadsheets, the relationship information has to be included manually once all the resources for the collection have been stored in the repository. To facilitate this process it is important to use appropriate titles for the resources that enable to identifying them easily, e.g. "Interview 1, teacher-1 date (transcript)". In the context of this application, analytical information - and relationship information - is key to enrich a qualitative collection, and the ways in which collection information can be displayed.

Element	Description	Example
isFormatOf	This relationship is useful when expressing that a resource is a different representation of another resource. This is particularly useful when describing the relationship between a transcript and an audio file. The transcript is a textual representation of the audio file	A transcript **isFormatOf** an audio file
isLinkedTo	Generic relationship to express that two resources are related	An image **isLinkedTo** a document
isVersionOf	This relationship is useful to relate the different versions of the same resource	draft 0.2 **isVersionOf** draft 0.1
isRelatedTo	More generic relationship to express that two resources are related	a Picture **isRelatedTo** a document
isAnnotationOf	This relationship is used to associate annotation with resources: memos with documents, code and categories with analysis documents, etc.	A memo **isAnnotationOf** a document
isReferencedBy	Relationship to express that a resource is referred to in another resource. This is useful when linking external publications to documents within a research collection	A publication **isReferencedBy** the research design document

CollectionManager

Collection Manager application

Updated Yesterday (21 hours ago) by amg...@gmail.com

QuDEx Collection Manager

The **QuDEx Collection Manager** application is also underpinned by a set of java libraries and its web interface makes use of the Spring Model-View-Controller (MVC) framework, and its Web Flow extension, to implement the **flows** of the web application - sequence of steps that guide the user to perform a series of operations over the digital repository. This application is concerned with the creation and storage of the documented datasets in the digital repository. It presents users with a stage-by-stage process that facilitates, firstly creating the collection structure in the repository, and secondly, integrating the documented files and their relationships within the created collection structure.

Overview

This system enables you to store files of many types - and links - into the Fedora repository together with appropriate descriptive metadata.

Prior to using this application, the different spreadsheets - based on the templates created with the Template Builder application - have to be filled and prepared with their associated zipped resources.

STEP 1: Select the type of operation: **Create new collection**, **Update existing collection** or **Add relationships**

STEP 2: Create NEW **Collection/sub-collection level** spreadsheet(s), with your choice of metadata fields, with the Spreadsheet Template Builder application

STEP 3: Complete Metadata spreadsheet and upload to the system

STEP 4: Create **File-level** spreadsheet(s), with your choice of metadata (documents in QuDEx terms)

STEP 5: Complete the different metadata spreadsheets and upload them into the system

CreateCollection

Creating a new collection in the repository

Updated Jun 4, 2013 by amg...@gmail.com

Introduction

A collection bundle comprises a set of spreadsheets for their upload into the fedora repository by using the Collections Manager Tool. Prior to using this tool, the user has to create both collection template spreadsheets and files template spreadsheets; fill them in with the relevant metadata information describing collections and files; and prepare the compressed files (preferably ZIP) containing the resources of the collection.

Structuring the collection

This is the first step when adding a collection to the repository. To perform this operation a set of collection spreadsheets is needed. The collection spreadsheet have to be uploaded in order, starting from the high level collection spreadsheet and descending down the collection tree. As an example, the following collection hierarchical organisation would produce the following set of spreadsheets:

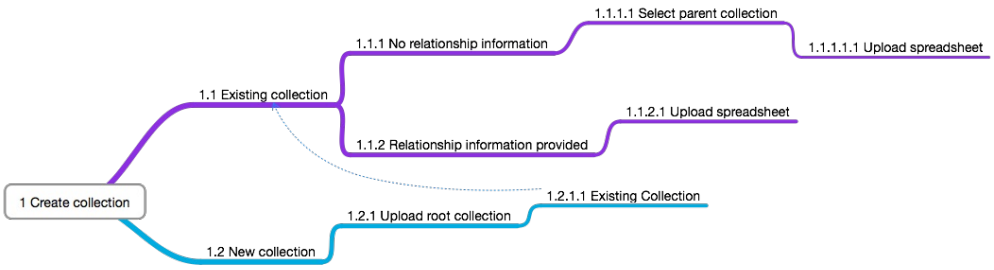
- Root collection
  - Sub-collection 1
  - Sub-collection 2
    - Sub-collection 2.1
    - Sub-collection 2.2

One spreadsheet describing "Root collection"; another spreadsheet with two entries describing "Sub-collection 1" and "Sub-collection 2"; and lastly, a third spreadsheet with two entries describing "Sub-collection 2.1" and "Sub-collection 2.2".

The process of uploading these spreadsheets is quite simple and requires repeating a number of steps per each level collection spreadsheet.

- Step 1. Select "Create new collection"
  - Step 1.1. New collection
- Step 2. Upload root collection spreadsheet. View screenshot
- Step 3. Select "Select existing collection" View screenshot
  - Step 3.1 The newly created root spreadsheet is selected.
- Step 4. Upload new level collection spreadsheet. View screenshot

Step 4 needs to be repeated as many times as the number of spreadsheets to upload.



Adding resources to the collection

Once the structure of the collection has been created in the repository by uploading the collections spreadsheets, the next step is to stored the resources associated with the collection by uploading the files spreadsheets along with the actual resources (in compressed files).

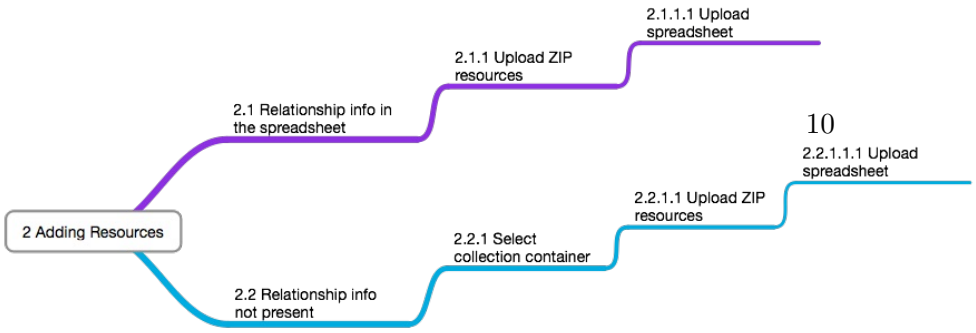
This process is very similar to the process of uploading collections spreadsheets, with the difference of the user having to upload and associate compressed files containing the collection's resources with the collection - or sub-collection - that will contain them.

- Step 1. Upload ZIP resources View screenshot
- Step 2. Select collection container
- Step 3. Upload files spreadsheets

View screenshot for Steps 2 and 3

If collection container information present in the spreadsheet

- Step 1. Upload ZIP resources
- Step 2. Upload files spreadsheets



UpdateCollection

Updating an existing collection in the repository

Updated Jun 4, 2013 by amg...@gmail.com

Updating, deleting collections

For simplicity, only collection level modifications are allowed, that is the documentation for collections and sub-collections can be updated; or collections deleted, in which case all the resources (files) associated with the deleted collection(s) would be purged. Files metadata cannot be updated or deleted directly.

In order to support collection and file level updates, the collection has to be deleted from the repository and the modified spreadsheet have to be uploaded again.

AddRelations

Specifying relationships between objects in a given collection

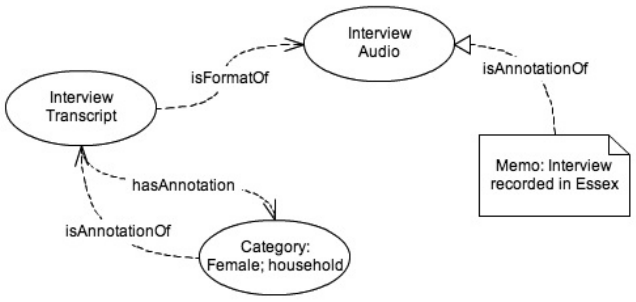
Updated Jun 4, 2013 by amg...@gmail.com

Introduction

Once the collection structure and the associated files or resources have been respectively created and stored in the repository, the collection resources can be enriched by adding semantic information about the relationships between them. This information includes a set of relationships included in the QuDEx schema vocabulary and are quite useful for the collection visualisation on the one hand, and the other if the information was to be shared or published in external content systems or other visualisation tools.

More details about the different types of relationships that can be specified are available here.

An example of the kinds of relationships that can be added to the resources in an existing collection is shown below.



Caption. A interview transcription is related to an audio file, which presents some memo annotations.

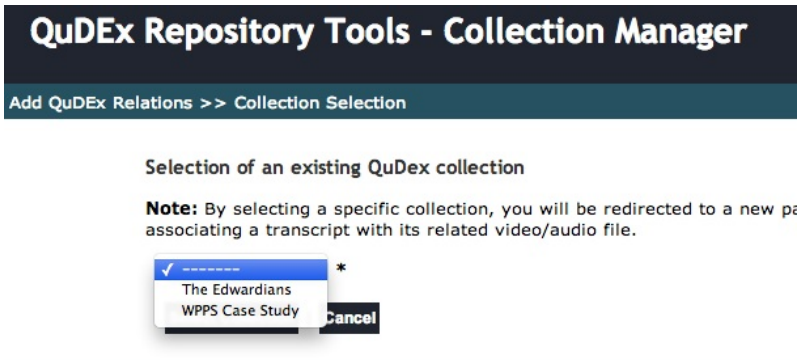
This information is included in the RDF database, via Fedora relationships API, in the form of triples - subject, predicate, object.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:qudex="http://example.org/qudex/">
        <rdf:Description rdf:about="info:fedora/qudex:interview">
              <qudex:isFormatOf rdf:resource="info:fedora/qudex:audio_int" />
              <qudex:hasAnnotation rdf:resource="info:fedora/qudex:category_int" />
        </rdf:Description> </rdf:RDF>
```

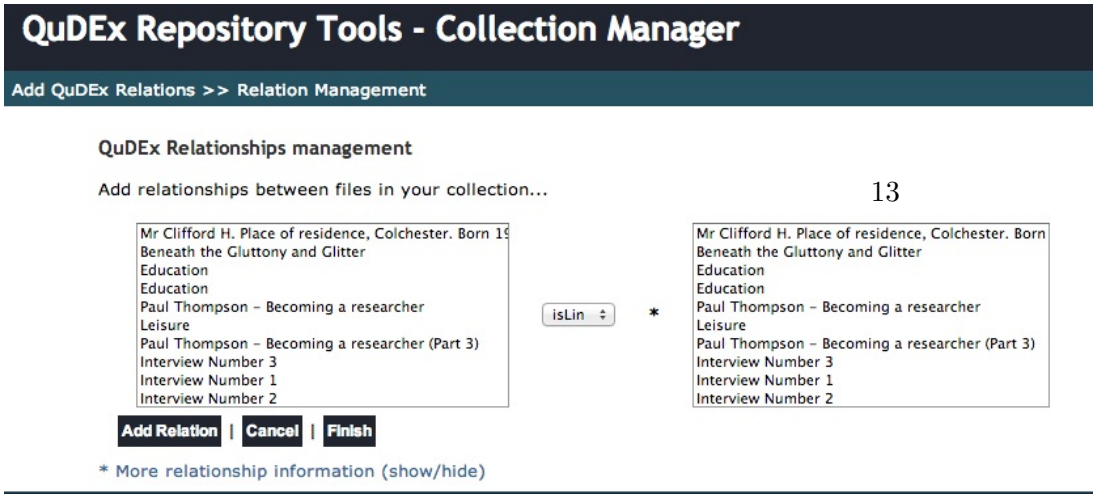
Caption. An RDF/XML representation of the relationships shown in the diagram above.

Process

The process of adding relationships between resources in a collection is quite manual at present, although it is simple enough. The user has to select the collection to which the relationships will be added:



Once the collection container is selected then the user is presented with a list of documents included in the collection which will be the source for the relationship; a list containing the different types of relationships that can be added; and a list of documents which will be the target of the relationship. The process can be repeated as many times as relationships to add.



QudexVisualiser
Collections Explorer application

Updated Sep 27, 2013 by amg...@gmail.com

QuDEx Collections Explorer

The **QuDEx Collection Explorer** relies on the Exhibit visualisation framework to implement a dedicated, faceted browser that enables to explore an individual digital collection. This application implements a set of java servlets that access the repository contents from the triplestore instance. The triplestore mirrors the contents - collection metadata and relationships information - of the repository and exposes them as RDF. The interface performs a set of SPARQL queries over the triplestore and the results are then processed by the Exhibit interface to generate the collection visualisation. More importantly, one of the advantages of mirroring the contents of the repository in the triplestore is that the data associated with a particular collection could also integrate relevant external information that is not included in the repository. Examples of such additional resources include citation information drawn from publishers' websites, or 'open data' from elsewhere on the web, and that is related to themes explored in the archived collection.

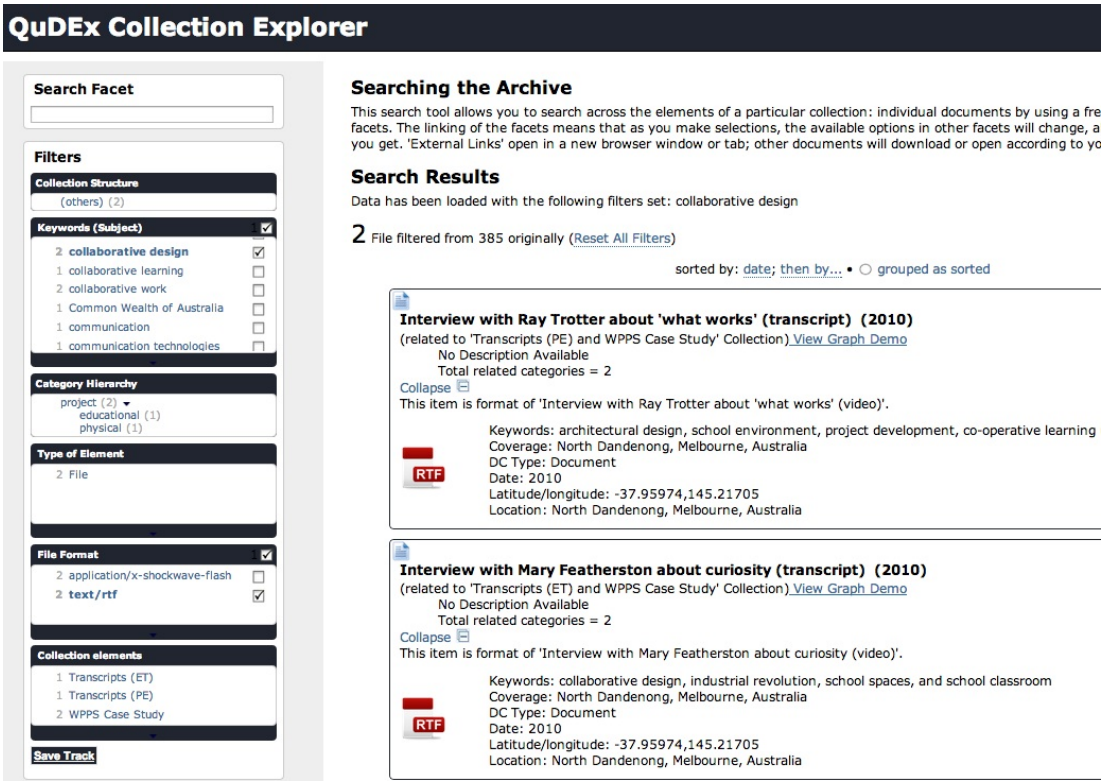
Note: this application has been tested with the following web browsers:

- Safari (6.0.5)
- Mozilla Firefox (23.0.1)
- Chrome (29.0.1547.76)

It is recommended not to use Internet Explorer since there are incompatibilities with the Javascript and JQuery libraries underpinning the Collection Explorer application. It will not visualise properly in Internet Explorer and this will also affect the functionalities of the application.

Overview

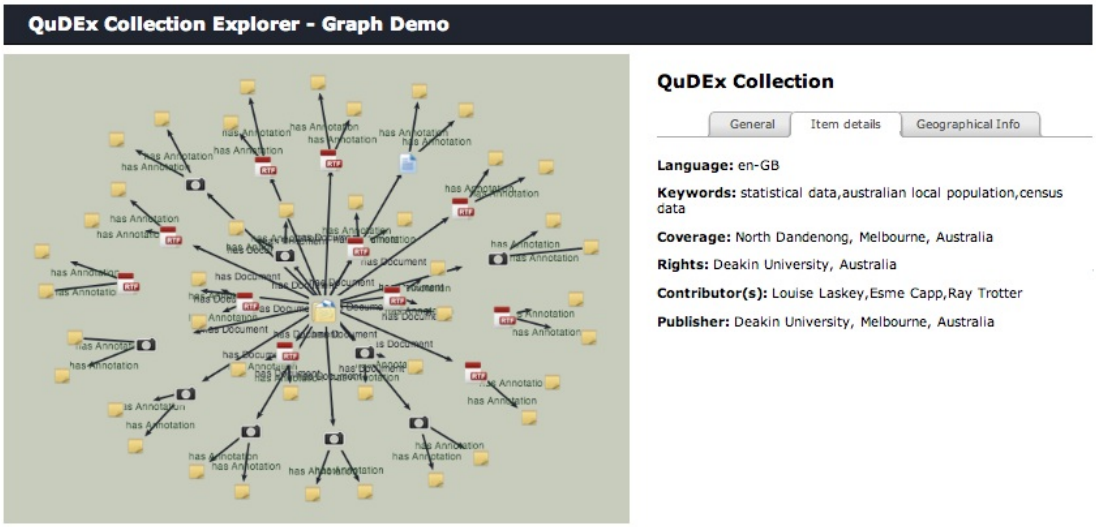
The collection explorer presents the users with a 'faceted' (property filters) navigation of the collection, by applying a set of filters. Faceted search enables users to explore a multi-dimensional information space - the collection structure, associated resources and analytical information such as codes, categories or notes/memos - by combining text search with a progressive narrowing of selections in each dimension. Each facet corresponds to a property or set of properties of the information elements included in the collection, that enable to filter the information by applying either an individual filter or a combination of them, similarly to 'AND' searches over a catalogue or a database. Given the nature of the information included in a collection, the most relevant facets that have been selected are the following: collection structure, which includes a hierarchical representation of the organisation of the collection; keywords, which lists all the related topics; category hierarchy, which includes the category framework with which the resources have been annotated; resource type and format; and `collection elements', which enables to explore the files by collection/sub-collection grouping. The way in which the user queries the data, by selecting the different filters, produces a reduced set of results that can be very useful when one is working with large datasets, e.g. one could visualise only the transcriptions associated with interviews that have been coded according to a selected set of topics.



Another functionality of the collection explorer is its ability to keep track of the user navigation, that is, the application can generate 'snapshots' of the search results after the user has applied some filtering to the dataset by simply generating an URL which contains a set of parameters to identify what filters have been applied. The latter allows the user to save the results of a set of queries over the dataset, which are then permanently accessible by simply using this URL in the web browser. This functionality could be quite useful to support the presentation of

results, similarly to 'enhanced publications' in which a publication is accompanied with research data, extra materials or post-publication data: student dissertations could be enriched if accompanied by the underlying data (evidence of the research); or online research diaries or research portfolios could reference to specific sub-sets of the archived collection by simply including a reference (URL) to a particular pre-filtered search. As the web pages generated using 'collection explorer' are entirely HTML based responses to a set of predefined queries, all that users need to do is construct that query to present their whole uploaded collection, a subset of that collection, or individual items in web pages, virtual learning environments, electronic portfolios or even online social environments. The aim is to make this process as simple and streamlined as the integration 'by URL' of a resource such as a Youtube video into a personal web page or social media stream like Twitter - but with the provenance and referential integrity of what is presented also being 'explorable' by other users and for secondary analysis.

Lastly, the collection explorer application incorporates a supplementary visualisation interface: the 'graph explorer'. This application presents a different approach to visualising the data from a particular collection. With the collection explorer, the user performs searches across an entire collection, by following a filtering and data reduction approach. In contrast, the graph explorer provides the user with a fine-grained, in depth exploration of individual sub-collection and file items that allows to visualise not only all the resources associated with a particular sub-collection but also it allows to explore how the elements relate to each other. The user can select any sub-collection in the collection explorer, and then is presented with an alternative graph visualisation showing the files, and their analytical annotations, that are included that collection. Alternatively, when the user selects an individual file from the collection explorer visualisation, he is presented with all the files - either belonging to the same sub-collection or across the entire collection - that are related to the previously selected file (see figure \ref{fig:4_7}). When selecting any element included in the graph, the user has then access to the element's metadata information and specific details, in a similar fashion to how the item information was presented in the collection explorer. Such a visualisation approach is possible owing to the definition and modelling of a set of relationships, both from Fedora's relationship ontology, to represent the structure of a study; and from the QuDEx qualitative schema, to represent all the relationships between files and their associated analytical elements. A person exploring a particular instrument from the collection, for example a transcription of an interview, is presented with methodological information such as the interview schedule used for that particular interview, a sub-set of the coded themes or notes associated with it, along with supplementary information such as policy documents or publications related to those identified themes.



DemoData

Toolkit Users: tutorial for creating a collection

Updated Yesterday (17 hours ago) by qudexrep...@gmail.com

Creating and storing a demo collection

The files needed to create a collection in the repository with the support of the QuDEx Collection Manager application can be downloaded here.

The demo-collection ZIP file contains the following data:

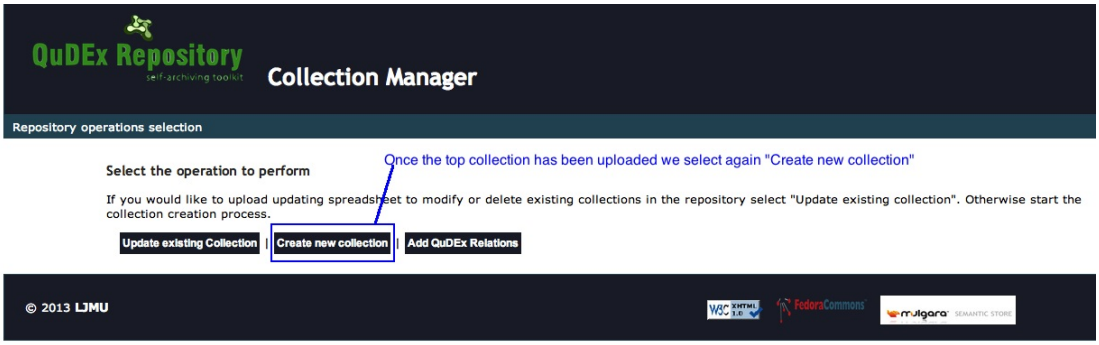
- **top-collection.xls**. This is the spreadsheet that contains the information of the demo collection. This is the first spreadsheet that needs to be uploaded into the QuDEx Collection Manager application.
- **sub-collection.xls**. This is the spreadsheet that contains the information of a sub-collection ("OS") of the demo collection. This is the second spreadsheet to be uploaded.
- **sub-collection-files.xls**. This is a "Files" spreadsheet which describes the resources that belong to "OS" sub-collection. Once the sub-collection has been uploaded, this spreadsheet has to be uploaded to store the resources in the repository.
- **filesshort.zip**. This is a compressed file that contains the physical resources that were described in the Files spreadsheet above mentioned. When uploading the Files spreadsheet in the repository, this ZIP file has also to be uploaded.

The structure of the demo collection in the repository is as follows:

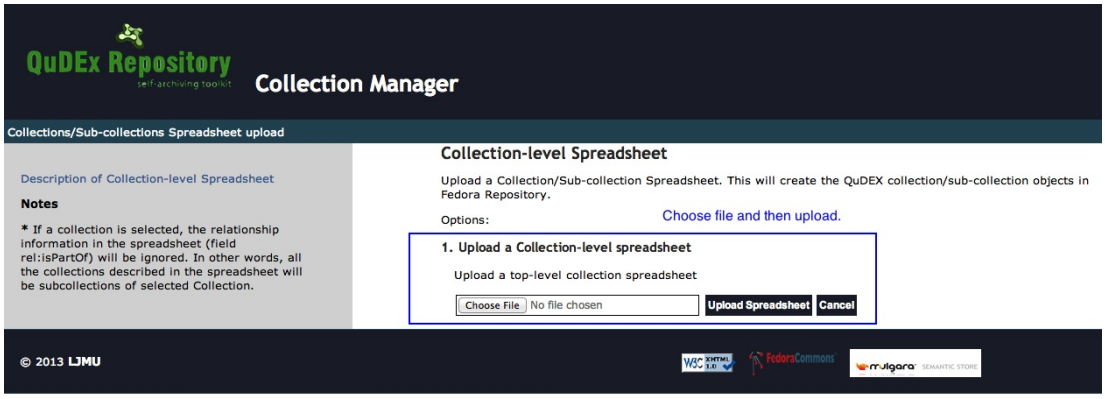
- Create a demo collection
- Add relationships to the elements of collection

Storing the collection (walkthrough)

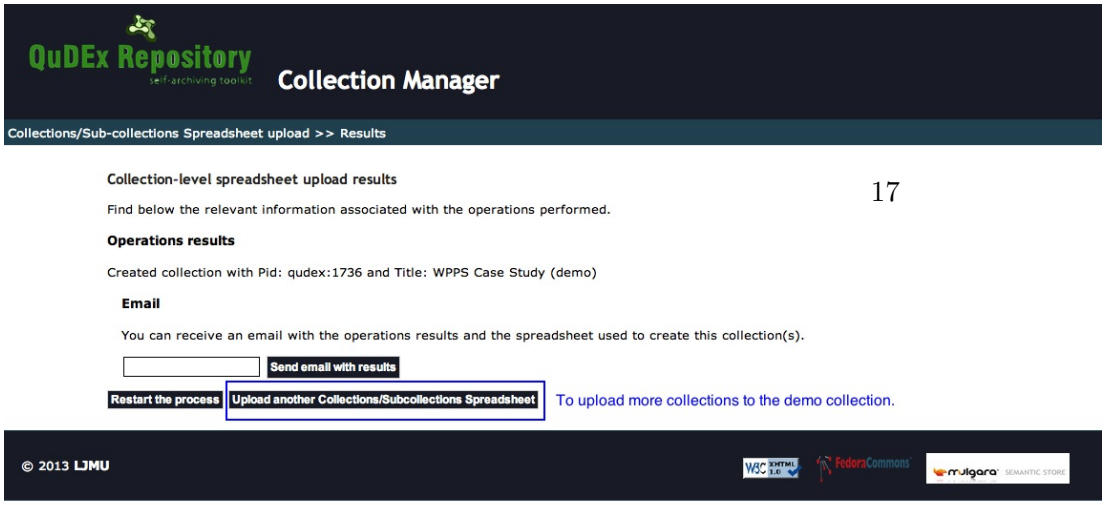
- Create new collection. From the Collection Manager application, first select "create new collection". Then, select "new collection".



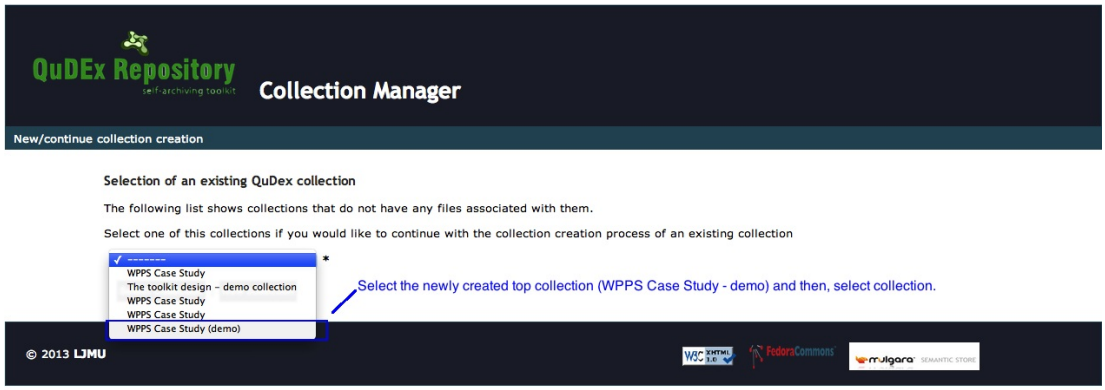
- Uploading the top collection. To upload the top level collection (top-collection.xls), simply choose the file and then hit the "upload spreadsheet" button.



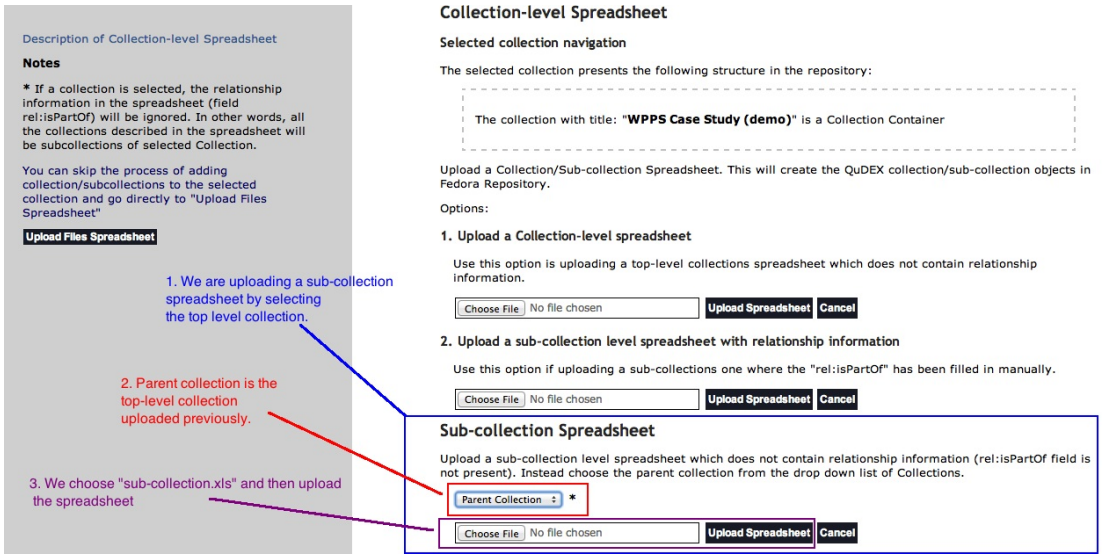
- Adding new collection results. To continue adding sub-collections or resources to the newly create demo collection, simply click "upload another sub-collection/collection spreadsheet".



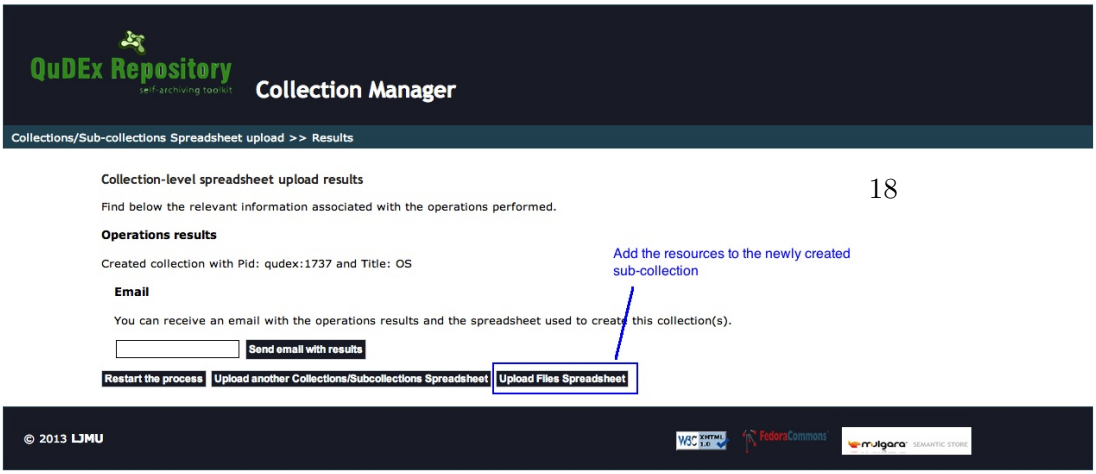
- Adding sub-collections. Once the top level collection has been uploaded, proceed to select this collection from the drop down list and then click on "select collection".



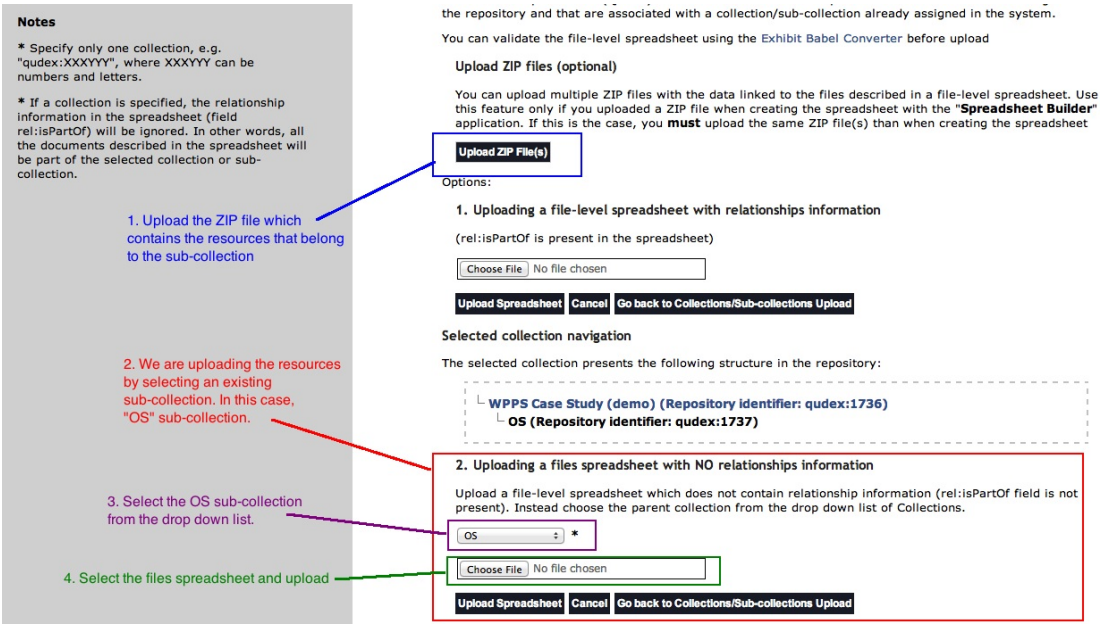
- Uploading the sub-collection spreadsheet. In this step, the sub-collection will be added by using the "Sub-collection Spreadsheet" section. Select "parent collection" from the drop down list and then choose "sub-collection.xls". Finally upload the spreadsheet.



- Adding sub-collection results. Once the collection structure has been created, select "upload Files spreadsheet" to upload the resources that belong to the newly created sub-collection.



- Adding resources to the sub-collection. First, upload the ZIP file (filesshort.zip) which contains the resources. Second, navigate to the section "uploading a files spreadsheet with no relationship information", and then select the newly created sub-collection (OS). Finally, upload the files spreadsheet.



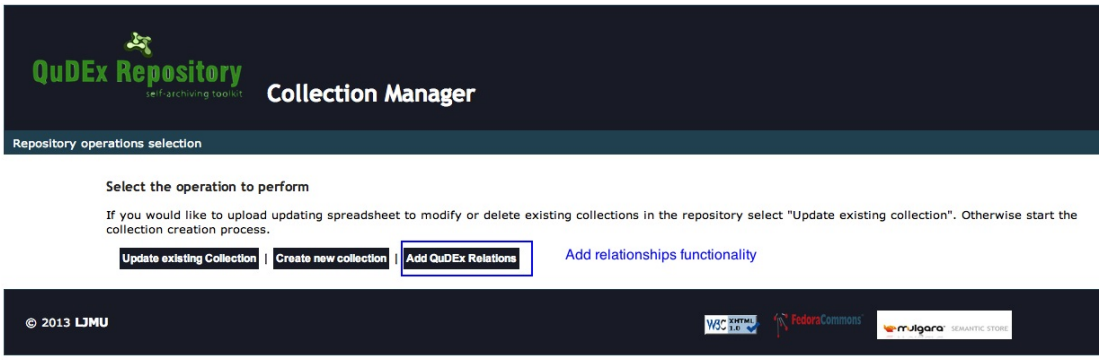
Tip: Write down the ID of the top level collection created in the repository. You will need this ID to visualise the collection with the QuDEx Collection Explorer application.

Adding semantic relationships

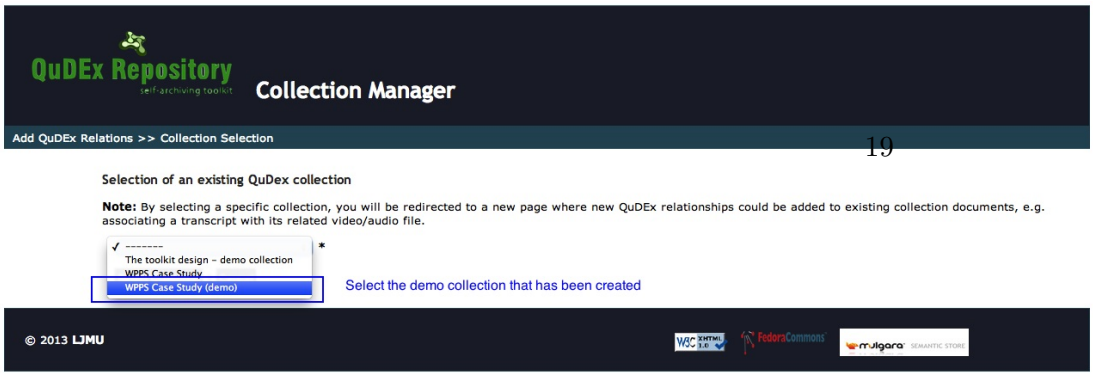
This demo collection represents a simple example of the kinds of data that can be created and stored with the support of the QuDEz Collection Manager application. Once the demo collection has been created, one can add more descriptive information about how the different elements included in a collection are related. This is explained in detail in this wiki page from the project's google code site.

Regarding this demo collection, we will add one relationship between the resources included in it, to demonstrate the Add relationships functionality. The demo collection has interview data, including an interview transcript and its associated audio clip. In this example, we will add a relationship to link the interview transcript with its associated audio so that it can be visualised in the Collections Explorer application.

- Add relationship information to a collection. Once the demo collection has been stored in the repository. Select **Add relations** from the main menu of the collection manager application.



- Select the collection. The next step is to select the collection to which we will be adding relationship information about its resources.



- Add a relationship. This demo collection only presents a sample of interview data, that is, a short clip and its associated transcript file. To add a relationship, select the transcript file first. Then, specify the type of relation (e.g. isFormatOf for interviews). Lastly, select the interview clip file and then click "add relation".