DSpaceOAISets

This page is now obsolete. DSpace 3.0 introduced a new OAI server written from scratch that supports virtual sets, filters and provides top performance. See OAI 2.0 Server.

DSpace OAI Sets

This page is related to the JIRA issue

The DSpace OAI offers only very basic sets out-of-the-box. Sets are always directly linked to DSpace collections. If customized sets are required, these can only be offered through collection mapping or (core) code changes. This requirement is essential for the DRIVER OAI harvester and some other OAI harvesters.

General side-notes:

- DRIVER states in its guidelines document that there is a requirement to name the set to be harvested "driver", but in practice a different set name can be used.
- DSpace support for deleted items is incomplete. The records will be returned when an item having only one collection (the owning collection) is withdrawn. They will not be returned if:
    - the item is expunged
    - the owning collection is changed and the harvested set was that previous collection
    - a mapping collection has been removed from an item and the harvested set was that removed collection
- In the OAI-PMH standard is no clear requirement on how removing a record from a set should be processed when only harvesting that set.
    - See also the question from Chris Wilper when developing Proai
    - To my understanding, at least a regular ListIdentifiers without a datespan is required when harvesting only one or a limited amount of sets

This page will review different solutions for offering sets.

Offering sets through collections

By default DSpace offers sets through collections. The item mapper could be used to determine which items need to be mapped to a given set.

Advantages:

- No code changes required
  Disadvantages:
- No option to choose the name of the set (requirement for e.g. DRIVER)
- Lots of manual labor for administrators

Offering sets through metadata

Alternative sets could be provided based on the metadata present in the item. This solution can be comparable to browse by subject where every subject would be a set and all items would have the different subject values as their respective sets (combined with the collections). The metadata fields to be mapped into sets needs to be configurable.

Advantages:

- Determining which items belong in a specific set can be done during submission or workflow.
- If the name of the set is important, the metadata value can simply be that exact set name.
  Disadvantages:
- No option to choose the name of the set (requirement for e.g. DRIVER)
    - There is a requirement to map the set name back to a specific metadatafield - metadatavalue combination (e.g. identifier.type_Article)
- Submitter or Metadata Editor will need to know what items need to offered for a given set.
- No support for combinations of metadata values
  Potential complexities involved:
- Deleted items need to be offered using these sets => regular browse indexes cannot offer this functionality

Offering OAI through software external to DSpace

There is open source software available specialized in offering OAI. Using this software can reduce the complexity for determining sets

Proai

Repository-neutral Java web application supporting OAI-PMH version 2.0

Proai can offer OAI based on a class implementing the OAIDriver. This implies DSpace is still responsable for listing the metadata formats, listing all sets and creating the XML response of each record. The metadata formats can be offered by DSpace using the oaicat.properties file. The sets can be the list of all collections, combined with the list of all metadata field - metadata value combinations to be offered as a set (see Offering sets through metadata). The XML response can be the result of the createMetadata() function combined with the listing of the sets which can be generated automatically from the list of all appropriate collections, combined with the list of all appropriate metadata field - metadata value combinations. This will however require a thorough rework of the DSpace OAI code.

Advantages:

- Set definition can be offered simply by defining the set in the xml. No need for DSpace to worry about offering only records within a specific based on e.g. browse indexes.
- *Might* better support changing collections on the items. If the owning collection is changed, or the item mapping is removed, the item should be marked as updated. More testing on whether this is supported is required
  Disadvantages:
- *Might* cause incompatibility with customized code already developed by individual institutions.
    ○ There is a requirement to offer proai a record xml through the print writer.
    ○ This might be possible to achieve without breaking the crosswalk by executing the crosswalk and sending that result to proai.
- Major redesign of DSpace OAI code

Virtual Sets (OAI extended addon)

At RCAAP project we developped an addon for DSpace that creates a virtual set based on filters and modifiers. This Addon was created to make DSpace repositories compliant with DRIVER guidelines, by defining a set driver to open access items.

The OAI Extended Addon is an addon (or a patch to be more precise) to the DSpace platform that extends the base functionality of the OAI-PMH interface already built into DSpace and delivers repository administrators more power and functionalities to select and filter the information they wish to expose to the outside world.

Initially the purpose of the OAI Extended Addon was to modify the OAI Interface's output, showing only items that were compliant with the DRIVER directives. Because, one of the primary characteristics of this addon is it's flexibility, the initial purpose, although being an important objective, degenerates in the OAI Extended Interface. The addon may be completely adjusted to other environments as its components can easily be configured, changed or extended, to respond to different information needs.

You can download the addon and the documentation here: OAI Extended Addon

Advantages:

- No code changes required
- Highly configurable

Feel free to ask more information or report bugs (info @ rcaap.pt)