Fedora OAI Provider

For information about the current status of OAI in the DLP, see the OAI page.

There are two standard OAI providers for Fedora. One is the built-in OAI provider and the PrOAI. PrOAI is a much more flexible solution. DLP is going to use PrOAI for OAI purposes.

The PrOAI provider

In Fedora 2.1 and on, the old OAI system still works, but the new system based on PrOAI is much more configurable. Of course, this means it is more difficult to configure. The PrOAI system periodically polls Fedora to update its "database" of deliverable records. This "database" is really a simple filesystem living under the proai.cacheBaseDir. The structure of this filesystem (and the associated tables in the relational database) is much like the way regular Fedora objects are organized.

Sample URL: http://fedora-dev.dlib.indiana.edu:8080/oaiprovider/?verb=ListRecords&metadataPrefix=oai_dc

Remaining issues (section last updated 10/04/2010)

- Some collections need to be imported into Fedora to be disseminated. These collections include VWWP and Wright.
- Metadata versions: There is a range of versions of metadata standards stored in our repository. Most of it consists of MODS versions 3.0 - 3.3. Until we find a way to deal with these, some of the records will not validate. This can be overcome initially by disabling metadata validation.
- Invalid metadata: Some of the metadata records in the repository do not validate mainly due to invalidly encoded characters. These records will need to be fixed. This is a rare case but disabling metadata validation might not help.
    - Furthermore, some of these items are valid, but the validator included in the oai provider incorrectly marks them as invalid
- The default socket time out value for resource index queries might be too low (which is set to 600 seconds by default in proai.properties).
    - Even when set to 3600 or an hour, this occasionally fails to return and ends up leaving fedora bogged down (sample log file)

Performance issues

- After the 6/24/09 deployment into production, the performance of fedora degraded until we had to kill the OAI provider.
- A new 4/13/2010 deployment resulted in the same behavior
    - looking at the logs showed a request every 12 minutes which was deduced to be the polling interval (120 seconds) plus the querySocketTimeout (600 seconds) suggesting that every 12 minutes an intensive resource index query was initiated and then abandoned 10 minutes later when it still hadn't returned.

Data questions

- Should we retain deleted hohenberger items and deliver them as deleted?
- Should we deliver records only for "cataloged" items?
- If items change status to a non-public status, should we deliver them as deleted?
- Should we deliver records for blocked items?
- Should we deliver records for copyrighted items?
- Should we deliver records for collections that aren't yet complete? (ie, we haven't approved the metadata for delivery)

Setup tips

Make sure you copy the JDBC driver into the OAI provider's lib directory.

The sample JDBC URL in the proai.properties file is misleading; just use the same one you're using for Fedora (assuming you have the proai user set up there).

The cache and sessions directories don't seem to work properly with directories that are on "virtual" drives (from the DOS SUBST command). It is probably easiest to use a relative directory name, starting with

```
webapps\\oaiprovider
```

to put it with the rest of the app in Tomcat.

Improper settings in the proai.properties, or manually messing with the cache directory, can sometimes corrupt the database. If the DB points to cached files that don't exist, the query process will seem to go through (no errors), but the result list will be empty. In this case, it is best to drop all of the associated DB tables (they start with "rc") and delete the contents of the cache directory, forcing a full rebuild the next time the OAI provider is started.

It will only attempt to index objects that appear to be oai items. An object appears to be an oai item if it:

1. Has an RDF property matching the configured driver.fedora.itemID. This ID will be used as the official "OAI ID".
2. Disseminates one or more of the formats specified in the Proai configuration.

If you want to index/provide objects based on a specific argument to a disseminator (like getMetadata?format=mods), the method must restrict its arguments to a defined set of values, and the Resource Index indexing must be set to level 2.

Sample RDF record (in RELS-EXT ds) for member objects:

```
<rdf:RDF xmlns:fedora="info:fedora/fedora-system:def/relations-external#"
  <!-- oai namespace added -->
  xmlns:oai="http://www.openarchives.org/OAI/2.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Description rdf:about="info:fedora/iudl:25959">
  <!-- oai itemID added -->
    <oai:itemID>oai:oai.dlib.indiana.edu:/inharm/sheetmusic/isl-xxx</oai:itemID>
    <fedora:isMemberOfCollection rdf:resource="info:fedora/iudl:23"></fedora:isMemberOfCollection>
  </rdf:Description>
</rdf:RDF>
```

Sample RDF record (in RELS-EXT ds) for collection (set) objects:

```
<rdf:RDF xmlns:oai="http://www.openarchives.org/OAI/2.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rel="info:fedora/fedora-system:def/relations-external#">
  <rdf:Description rdf:about="info:fedora/iudl:23">
    <!-- oai setSpec and setName added -->
    <oai:setSpec>isl</oai:setSpec>
    <oai:setName>InHarm - ISL collection</oai:setName>
  </rdf:Description>
</rdf:RDF>
```

See OAI Data Provider Requirements for some more setup tips.

Note about Metadata validation

PROAI can validate XML during the indexing process. But there is no documentation on how it is configured and several attempts at indexing with validation turned on failed. I'm temporarily turning off validation. After we figure out how it is configured we might want to turn it back on.

Indexing process

When proai.service.ProviderServlet starts,

1. It loads the proai.properties file
2. It creates a Responder based on the properties, which
    a. Creates a RecordCache, which
        i. Initializes the OAIDriver specified in the properties file (in this case the FedoraOAIDriver)
        ii. Starts a thread that periodically runs OAIDriver.listRecords
            1. FedoraOAIProvider.listRecords passes off to
                a. ITQLQueryFactor.listRecords, which
                    i. sends date-constrained queries to the Fedora Resource Index.
                    ii. creates a CombinerRecordIterator, which
                        1. creates a FedoraRecord for each result item
        iii. calls FedoraOAIProvider.writeRecordXML
            1. This method name has been changed through a different version of PrOAI, but it almost definitely hooks up to FedoraRecord.writeMetadata

Query process

Queries to http://localhost:8080/oaiprovider/? go through proai.service.ProviderServlet.

A sample query for ListRecords:

1. proai.service.ProviderServlet gets verb=ListRecords&metadataPrefix=oai_dc
2. ProviderServlet.doGet passes to Responder.listRecords
    a. which passes to SessionManager.list
        i. which spawns a new CacheSession (a thread), and returns its getResponseData (this method blocks until the thread has completed)
            1. fills output files (under the session directory, one file per response page) with the paths of cached metadata files that meet the criteria

The Fedora-based OAI provider

In Fedora 2.0, OAI export is built in, and works without any additional configuration.

For example, see:

- A list of items
- OAI-DC record for one item

However, this system has two major drawbacks:

- it only delivers DC records
- it delivers a record for every item in the repository, regardless of its type

Moving Existing Collections into Fedora OAI

This page lists what needs to be fixed before moving the existing collections into Fedora OAI.

- Cushman Collection metadata has been exported using the existing OAI provider and ingested into production Fedora as is (i.e. some metadata fix might still be needed). See this example object METS: http://fedora.dlib.indiana.edu:8080/fedora/get/iudl:32066/METADATA