OAI Data Provider Requirements

OAI Data Provider Requirements

A new, Fedora-based OAI data provider for use in the IU Digital Library Program must implement the following optional parts of the OAI protocol. See the protocol documentation and implementation guidelines for more information on these features.

Each of the requirements is augmented with information about its support in the Fedora OAI Provider, denoted **FOP**.

1. Support for sets, including the potential for a single item to appear in multiple sets
   - **FOP**: This is handled by RELS-EXT isMemberOf relationships.
2. Must support delivery of "deleted" records
   - **FOP**: It works, but changes to object states don't take effect very quickly (do they at all?). Rebuilding the oaiprovider cache fixes this. The log files seem to indicate that the state is based on each datastream/disseminator that is used by the provider, so these may all need to have their state changed.
3. Support for multiple metadata formats for a single item
   - **FOP**: easily configured, provided there is a datastream/disseminator for the desired format
4. Support for multiple versions of the same metadata format within the repository (i.e., one set in MODS 3.0, one set in MODS 3.1, others in MODS 3.2), but not necessarily multiple versions of the format within a single object
   - **FOP**: Can deliver anything Fedora can hold.
5. Support for the Repository-level <description> container
   - **FOP**: Yes, supports arbitrary Identify information.
6. Support for the Set-level <setDescription> container
   - **FOP**: Yes, stored in a special datastream of the set object.
7. Support for the Record-level <about> container
   - **FOP**: As long as there is a datastream/disseminator for it.
8. Must support arbitrarily including/excluding items from the provider.
   - **FOP**: The only way to prevent something from being indexed is to not give it an ID or a metadata datastream. It seems best to handle this by only moving items to the production Fedora instance when we are ready to disseminate them via OAI.
9. Configurable set names
   - **FOP**: Yes. Depends on the name of the object stored in the relationship.
10. Configurable set membership (e.g., don't assume a set is always a DLP collection)
    - **FOP**: Yes.
11. DESIRED: Support for <friends> description container at the repository level.

Open questions

1. At what level do we need to include/exclude objects? At the collection level only? This affects what kind of flag we use to indicate that an item can be indexed. *JLR: For collections that grow over time we may want to have item-level control over whether something is exposed via OAI, to allow for exposure only once it's "ready" (whatever that means).*