

Metadata Support Priorities

The priorities below are based on the [October 2011 community survey on improving metadata support](#) developed by the [DSpace Community Advisory Team](#) at the request of the [DSpace Committers/Developers](#). DCAT has summarized and interpreted the survey results and listed the community priorities below based on the estimated level of effort required. DSpace community members should feel free to add more detailed explanation/examples to further flesh out the changes/requirements they would like to see. Community project teams can be formed around any of the priorities. If you would like to participate or lead any of the project teams, please contact Valorie Hollister at vhollister@duraspacespace.org.

1) Updating the Qualified Dublin Core registry in DSpace to the latest standards of the DCMI

LEVEL OF COMMUNITY EFFORT: likely low to medium

LEVEL OF DEVELOPER EFFORT: likely low to medium

- Qualified Dublin Core registry in DSpace has not been updated for 8 years. Standardization would help compatibility with other systems and benefit from information gathered by the DCMI community (related links: <https://jira.duraspace.org/browse/DS-805> <https://jira.duraspace.org/browse/DS-433>).
- need consensus/guidance from metadata experts in the community on specifically how/what to update and which version of the DCMI standard should be used
- need to link to the relevant schema on DC website and lock down (allow adds, but not changes)
- isolate where local customizations are done
- [Initial look at the requirements](#)

2) Standardizing the default namespaces

LEVEL OF COMMUNITY EFFORT: likely low to medium

LEVEL OF DEVELOPER EFFORT: likely low to medium

- Currently instead of creating a customized metadata schema, some DSpace repository managers edit the default registry, effectively breaking compliance with the standard Dublin Core. This can create a problem for the portability of data to/from of your repository. It has been proposed that in the future that DSpace would include 3 different metadata schemas, to insure that the metadata will be easily portable to other systems:
 - 1) standardized default Dublin Core, un-editable (no changing or deleting fields) except to add fields with qualifiers to existing elements
 - 2) a yet-to-be developed DSpace admin/internal metadata fields that describe DSpace specific fields
 - 3) an empty local, easily customizable schema
- need to validate the above strategy and confirm that if it was implemented it would prohibit changes and deletions that would break compliance with the standardized default Dublin Core metadata schema
- **?Citation metadata?:** perhaps use a subset of PRISM for citation metadata include it as a new namespace *IS THIS THE RIGHT CATEGORIZATION FOR THIS OR SHOULD IT GO UNDER #5?*

3) Adding metadata authority controls/vocabularies to the data model

LEVEL OF COMMUNITY EFFORT: likely medium to high

LEVEL OF DEVELOPER EFFORT: likely medium to high

- Although you can manage the metadata scheme in the DSpace data model, there is little out-of-the-box support for vocabularies (like DCMI Type) and encoding schemata.
- authority controls exist already, need to identify what specifically the community needs/what will work better
- develop educational materials/tutorials for the community about the current add-on
- not allowed to ship DSpace w/Library of Congress headings
- update challenge - need to open up rights to use a controlled vocal and link from an external source so it doesn't have to be brought into DSpace
 - example: Nat'l Agricultural Library - linked open data, LC, subject based, NLM

4) Moving metadata related configurations from dspace.cfg to the database

LEVEL OF COMMUNITY EFFORT: likely medium to high

LEVEL OF DEVELOPER EFFORT: likely medium to high

There are currently no verifications or safeties when editing/removing metadata fields in your schemas that are used in dspace.cfg for configuring browse parameters, search indexes, visibility (on simple and full item view). If this functionality was moved to the database, not only would it create a webUI config and hints, it would also provide improved accessibility for the repository managers.

5) Develop support for additional metadata standards (use the "Other" field to specify which standards)

LEVEL OF COMMUNITY EFFORT: likely medium to high

LEVEL OF DEVELOPER EFFORT: likely medium to high

- depends on what schema (XML based schemas are more complex - dspace doesn't support XML - MODS,
- MODS implementation better done after fedora intergration since it is XML
- need metadata expert
- **Hierarchal metadata:**
 - add functionality / move towards relational aspect functionality - like MODS
 - allow relationships between community/collection/bitstream metadata
 - possible iterative solution - create functionality that emulates hierarchal behavior
 - example: adding in author affiliation
 - example: metadata on bitstream - related back to items
- **?Citation metadata?:** perhaps use a subset of PRISM for citation metadata include it as a new namespace *IS THIS THE RIGHT CATEGORIZATION FOR THIS OR SHOULD IT GO UNDER #2?*

6) Improved (or more transparent) flexibility:

LEVEL OF COMMUNITY EFFORT: likely high

LEVEL OF DEVELOPER EFFORT: likely high

- simplify/make local customizations more accessible through UI:
 - simplify where/how things are done - bring more functionality to the UI or the metadata registry
 - create more educational/tutorials on how/where to do things (i.e. how to plug in metadata schemas)
 - import/export to metadata other than DC (like in Eprints where you pick a format for export)
- expose RDF triples

7) Enhancing the metadata available for Communities, Collections and Files (bitstreams)

LEVEL OF COMMUNITY EFFORT: likely high

LEVEL OF DEVELOPER EFFORT: likely high

- Unlike DSpace Items, DSpace Collections, DSpace Communities and DSpace Files (bitstreams) currently do not have Dublin Core metadata associated with them, making the available descriptions much less granular.
- get discussion started and find some dev interested in architecting it
- very large changes, every single object in DSpace, will present and upgrading challenging to convert to an entirely different object model could be preparation for integration with Fedora - re-factoring how best we can re-work dspace with support in the next few versions - not a quick jump