

Proposal to Update DC Registry and Add DCTERMS Registry

- [Introduction](#)
- [Glossary of Terms](#)
- [Main goal of these recommendations](#)
- [Possible Phases of Update](#)
- [Outstanding issues for committers and community](#)
- [Recommendation background](#)
- [Relevant JIRA tickets](#)
 - [Would be RESOLVED](#)
 - [RELATED/Would be AFFECTED](#)
- [Areas/processes that will be affected by registry update](#)

Introduction

Recommendations towards "Updating the Qualified Dublin Core registry in DSpace to the latest standards of the DCMI," a priority identified in the October 2011 community survey on improving metadata support. It also seeks to comply with the proposal to [Standardize the Default Namespace](#).

Note: In addition to the child page of mappings linked below, see the grandchild pages, "[Samples and decision points for mappings.](#)" and "[Proposed phased schemas](#)"

Glossary of Terms

- DCMI
 - [Dublin Core Metadata Initiative http://dublincore.org/](http://dublincore.org/)
 - The Dublin Core Metadata Initiative (DCMI) manages the long term curation and development of DCMI [specifications and metadata terms namespaces](#)
- DCMI Metadata Terms
 - <http://dublincore.org/documents/dcmi-terms/>
 - Specification of all metadata terms maintained by the Dublin Core Metadata Initiative, including properties, vocabulary encoding schemes, syntax encoding schemes, and classes
- DCTERMS ('dcterms')
 - Properties in the /terms/ namespace <http://dublincore.org/documents/dcmi-terms/#H2>
- DC
 - 'Simple Dublin Core' (elements only, no qualifiers)
 - Properties in the /elements/1.1/ namespace <http://dublincore.org/documents/dcmi-terms/#H3>
 - DCMES - Dublin Core Metadata Element Set, Version 1.1 <http://dublincore.org/documents/dces/>
 - The fifteen terms of the Dublin Core Metadata Element Set, which have also been published as IETF RFC 5013 [[RFC5013](#)], ANSI/NISO Standard Z39.85-2007 [[NISOZ3985](#)], and ISO Standard 15836:2009 [[ISO15836](#)]
 - 'dc' - Schema name of the DSpace (up to 4.x) metadata registry default/shipped with schema (incorrectly referred to as dcmi-terms)

Main goal of these recommendations

- The ultimate goal of these recommendations is to implement DCTERMS as the default metadata schema, thus ensuring compliance with current standards endorsed by DCMI.

Possible Phases of Update

Ultimate goal = 'dcterms' schema as metadata registry default schema

For proposed phased schema changes see: "[Proposed phased schemas](#)"

Phase One: Add schema "dcterms" to the DSpace 4.0 default registry as an added schema (not the default schema)

- Schema "dcterms" is added to DSpace 4.0 for testing.
- Provide documentation.
- Curation task for metadata migration added to 4.0.

Phase Two: Change default schema from "dc" to "dcterms". The "dc" schema is updated. "local" and "dspace" schemas are added to default registry

- Develop and implement flat "dcterms" schema as the default DSpace schema.
- Develop and implement "local" schema.

- Develop and implement a DSpace admin/internal metadata schema - "dspace"
- Ship DSpace with "dc," "dcterms," "dspace," and "local" schemas in metadata registry.
- Create new Application Profile for DSpace.
- Address all areas of code affected (search/browse, import/export, crosswalks, hard-coded 'dc' elements/qualifiers, etc.). Resolve issues with features that rely on metadata solutions (i.e., Creative Commons, RequestCopy, embargo, etc.). Consider plugins, add-ons, interfaces, etc. as well.
- Lock down the "dcterms" schema from the UI.
- Provide tools for current DSpace repositories to migrate to these schemas (i.e. edit their metadata registry and data), if desirable (i.e. provide tools for migrating elements not compliant with dcterms to "local" registry).
- DSpace add-on for remote query to profile the metadata usage of DSpace repositories added.
- Provide documentation.

Outstanding issues for committers and community

- Is it possible to ultimately implement DCTERMS with full functionality (vocabularies, etc.)? What changes to the data model will be necessary?
- How will this proposal integrate with other suggested changes to DSpace metadata, including [Proposal for Metadata Enhancement](#)? How might it affect integration with Fedora? How might it affect other desired changes to metadata in DSpace, including implementing functional structured metadata such as MODS, METS, and PREMIS?
- What challenges will this proposal present—or solve—for **harvesting**?
- To enable repositories to migrate existing metadata to the DCTERMS schema, we will need to develop robust tools for repositories to deploy. (Note: A curation task has been added to 4.0.)
- Should DSpace admin/internal metadata (not including DIM) have its own schema ("dspace"), or use 'local' schema?

Recommendation background

The original DCAT Discussion forum topic that lead to this proposal can be found at ["Updating the Qualified Dublin Core registry in DSpace."](#)

- Update current default 'dc' schema in DSpace metadata registry to current standards
 - *Background:*
 - The default DSpace metadata registry ships with the 'dc' schema (which is the default DSpace metadata schema). It was designed to comply with the Dublin Core Libraries Working Group Application Profile, modeled on flat, extensible Qualified Dublin Core.
 - [The default DSpace metadata registry schema is "namespace" = <http://dublincore.org/documents/dcmi-terms/> and "name" = dc. This use of "dcmi-terms" should not be confused with [DCMI Metadata Terms](#). It is a qualified 'dc' schema for DSpace based on an application profile (Dublin Core Libraries Working Group Application Profile (LAP)). DSpace used the LAP as the starting point for its application of Dublin Core, borrowing most of the qualifiers from it and adapting others to fit. Some qualifiers were also added to suit DSpace needs. The 'namespace' it is declaring is not a DCMI namespace. The default DSpace schema is not dc: namespace <http://purl.org/dc/elements/1.1/> (the collection of legacy properties that make up the Dublin Core Metadata Element Set, Version 1.1 [DCMES]) or dcterms: namespace <http://purl.org/dc/terms/> (the collection of all DCMI properties, classes and encoding schemes (other than the properties in the Dublin Core Metadata Element Set, Version 1.1 [DCMES], the classes in the DCMI Type Vocabulary [DCMI-TYPE] and the *terms* used in the DCMI Abstract Model) <http://purl.org/dc/terms/>)]
 - *Desirability of and demand for DCTERMS compliance:*
 - [Updating QDC to the latest standard of DCMI](#) and [Standardizing the default namespaces](#) were identified as priorities in the October 2011 community survey on improving metadata support authored by DCAT at the request of the DSpace Committers /Developers. These updates were seen as standardizations that ["would help compatibility with other systems and benefit from information gathered by the DCMI community"](#) and would address [the problem of broken compliance with the standard Dublin Core](#), which had affected portability to/from a repository.
- Add DCTERMS as new, parallel schema in the default metadata registry
 - *Background:*
 - DCMI has not updated its Qualified Dublin Core standard since 2005. The community standard has shifted towards [DCMI Metadata Terms](#), which, unlike QDC, is not a flat schema based on the schema.element.qualifier format. DCTERMS include range and domain values. A particular term may link to another term that it refines or is refined by (for example: the dcterms "hasPart" refines "relation"; "created" refines "date").
 - *Rationale:*
 - DCTERMS is the currently maintained DCMI standard.
 - As Sarah Shreeves [recently commented](#):
"I want to strongly urge the group to look at conforming with DCMI terms (<http://dublincore.org/documents/dcmi-terms/>) - even if we can't conform to the vocabulary, etc, this is the most up to date and current form of the namespace. If we use the dc qualifiers document we will be perpetuating the same problem, IMO. I think we can, as Tim suggests, have a graceful path forward. I will admit that a real part of my fear of just moving to DC Qualified is that DSpace--in terms of metadata--will continue to be seen as out of touch with where much of the metadata world is headed."
 - Also, from <http://dublincore.org/documents/dces/>:
"Since 1998, when these fifteen elements [dc: namespace] entered into a standardization track, notions of best practice in the Semantic Web have evolved to include the assignment of formal domains and ranges in addition to definitions in natural language. Domains and ranges specify what kind of described resources and value resources are

associated with a given property. Domains and ranges express the meanings implicit in natural-language definitions in an explicit form that is usable for the automatic processing of logical inferences. When a given property is encountered, an inferring application may use information about the domains and ranges assigned to a property in order to make inferences about the resources described thereby. Since January 2008, therefore, DCMI includes formal domains and ranges in the definitions of its properties. So as not to affect the conformance of existing implementations of "simple Dublin Core" in RDF, domains and ranges have not been specified for the fifteen properties of the dc: namespace (<http://purl.org/dc/elements/1.1/>). Rather, fifteen new properties with "names" identical to those of the Dublin Core Metadata Element Set Version 1.1 have been created in the dcterms: namespace (<http://purl.org/dc/terms/>). These fifteen new properties have been defined as subproperties of the corresponding properties of DCMES Version 1.1 and assigned domains and ranges as specified in the more comprehensive document "DCMI Metadata Terms" [DC TERMS]. Implementers may freely choose to use these fifteen properties either in their legacy dc: variant (e.g., <http://purl.org/dc/elements/1.1/creator>) or in the dcterms: variant (e.g., <http://purl.org/dc/terms/creator>) depending on application requirements. The RDF schemas of the DCMI namespaces describe the subproperty relation of dcterms: creator to dc: creator for use by Semantic Web-aware applications. Over time, however, implementers are encouraged to use the semantically more precise dcterms: properties, as they more fully follow emerging notions of best practice for machine-processable metadata."

- Lockdown schemas offering migratory tools to pull out local customizations and push into new local schema. Make it possible but not easy to delete or edit elements in DCTERMS schema. Continue to enable the addition of qualifiers in the 'dc' schema.
- For staging purposes, we recommend that DSpace ship with 4 registries in Phase 2, to support ultimate migration to DCTERMS and to standardize namespaces by pushing local customizations not compliant with DC or DCTERMS into a local schema.:
 - 1) 'dcterms' (DCTERMS) - which will be the default metadata schema
 - 2) 'dc' schema
 - 3) 'dspace' schema for system/admin metadata
 - 4) 'local' schema - which would ship with some elements migrated out of 'dc' because not compliant with QDC, and enabled for the purpose of local customizations

Relevant JIRA tickets

(please add any JIRA tickets that could be affected by this proposal!)

Would be RESOLVED

[DS-433](#) - Getting issue details...

STATUS

[DS-805](#) - Getting issue details...

STATUS

[DS-716](#) - Getting issue details...

STATUS

RELATED/Would be AFFECTED

[DS-125](#) - Getting issue details...

STATUS

[DS-202](#) - Getting issue details...

STATUS

- Not sure whether this is related. But I assume if DCMI requires some properties to be unique (for example identifiers), I guess you would need a generator to ensure unique identifiers get generated.

[DS-651](#) - Getting issue details...

STATUS

[DS-716](#) - Getting issue details...

STATUS

[DS-800](#) - Getting issue details...

STATUS

[DS-815](#) - Getting issue details...

STATUS

[DS-1134](#) - Getting issue details...

STATUS

Areas/processes that will be affected by registry update

What areas and processes will be affected by these shifts? Is there any documentation of what features in DSpace are making use of certain fields? Where will the code be affected? Where are metadata elements hardcoded?

(pulled from [September 4, 2012, DCAT discussion](#))

- Any processes that create new metadata in DSpace:
 - submission forms
 - spreadsheet importer
 - command line import
 - SWORD
 - built-in OAI Harvester
- Any process that displays metadata in the web user interface:
 - item pages
 - search, browse, DSpace discovery
- Any process that delivers the metadata (potentially via crosswalks) to other applications:
 - OAI server
 - REST API