Scholaris Metadata and Discovery Expert Group (S-MDEG)

Minimum Viable Metadata Recommendations

Release 1.1

Draft for review, July, 2026

Scholaris Metadata and Discovery Expert Group (S-MDEG)

Co-chairs

- Robin Desmeules (McGill University)
- Mike Nason (University of New Brunswick)

Committee Members

- Caitlin Bakker (University of Regina)
- Teresa Bascik (Université de Montréal)
- Carl Brouillard (Université de Sherbrooke)
- Pascal Calarco (University of Windsor)
- Sharon Farnel (University of Alberta)
- Laura Smart (Western University)
- Carolyn Sullivan (University of Ottawa)

Former Members

- Jordan Pedersen (University of Guelph) - former co-chair

Subgroup Members

- Teresa Bascik
- Carl Brouillard
- Pascal Calarco
- Sharon Farnel
- Mike Nason

Related recommendations can be found in the Scholaris Network of Experts Reports collection.

www.carl-abrc.ca

CARL ABRC
CANADIAN ASSOCIATION OF RESEARCH LIBRARIES
ASSOCIATION DES BIBLIOTHÈQUES DE RECHERCHE DU CANADA

Table of Contents

Introduction 2
Minimum Viable 2
About Dublin Core 2

Recommended Minimum Metadata Elements and Scope Notes 3
Generic Dublin Core for DSpace 4
- Contributor Metadata 4
- Date Metadata 4
- Rights (Access/Licenses) 4
- Related Works and Citation Metadata 5

OpenAIRE Compliant Spec 7
- Contributors 8
- Rights (License) 9
- Related Works and Citation Metadata 9

Theses & Dissertations / LAC Spec 14

Threading the Needle: Cross-walks and Implementation of OpenAIRE/ETD-MS/Rioxx or Other Schema 14

Controlled Vocabularies 17
Access Rights 17
Disciplines / Subjects 17
Language (dc.language.iso) 17
Type (dc.type) 18
Version Types 18

Ambitions Outside Scope 18
Date Format 18
Collections Hierarchy 19

On Responsible Custom Metadata 19
Custom Qualifiers 20
Custom Namespaces 20
Examples 21
- Example 1. Custom Qualifier Replacing Faculty/Department 21
- Example 2. Custom Qualifier to Accommodate Migration from MODS. 21
- Example 3. Custom Property to Capture Thesis/Dissertation Specialization 22
- Example 4. Using Existing LOD Vocabularies to Capture Degree, Department, and Faculty 22

Introduction

The Minimum Viable Subgroup – a subset of the larger Scholaris Metadata and Discovery Experts Group (S-MDEG) – is tasked with creating recommendations for minimum, standard metadata within the Scholaris service. Broadly, this means clear-cut documentation for standard, Qualified Dublin Core as utilized by DSpace >7. Additionally, this subgroup intends to also identify metadata mappings appropriate for additional namespaces, including OpenAIRE, thesis/ETD-MS, and basic custom metadata recommendations.

The recommendations herein are meant to align with existing schema, best practice, minimal conflicts, and default DSpace metadata crosswalks. Custom metadata or variations on the recommendations herein are certainly inevitable. However, any deviations from default behaviour of DSpace requires your metadata adhere to the following as much as possible:

- metadata should be recorded consistently across all items in your repository,
- metadata should be cross-walked responsibility where possible/relevant,
- metadata – in particular, local deviations, applications, or interpretations – should be documented internally in your metadata registry (e.g., in a Scope Note), and
- where possible, metadata should be machine readable (stick to relevant controlled vocabularies and URL/URI).

Version 1.1 Update, June 2026

This document was updated in the Spring of 2026 with a small number of updates and adjustments for clarity, compatibility, and expressed frictions from the original release. Summary of changes includes:

- addition of recommendations/resources for deposit forms
- added clarity for relational metadata and potential compromises in Dublin Core
- added clarity for rights (access & license) language in light of upcoming deprecations and a change to metadata behaviour in DSpace 8
- added recommendations for funding metadata

Minimum Viable

For the purposes of this group, "minimum viable" depends on the prospective destinations for metadata and consists of descriptive elements and qualifiers which are essential for optimizing data quality and information retrieval within an aggregated metadata environment and ensuring conformance with international standards for scholarly repositories.

Repository metadata needs to be usable externally to the institutions that host them. As a result, this document contains three discrete tiers of metadata recommendation. The first is simple, qualified Dublin Core with an eye to where standard metadata should live in DSpace. The second incorporates the OpenAIRE namespace, which offers more granular metadata than Dublin Core. OpenAIRE is not a requirement of Scholaris, but successful harvesting of a repository to the service can significantly improve discovery of repository materials. The third tier is specific to Theses & Dissertations with an eye to valid metadata, ETD-MS, and successful ingest by Library & Archives Canada.

It is worth noting **that there is no single, frictionless recommendation for metadata in this space**. Accommodation of two disparate external schema, coupled with pre-existing standards, interpretation *of* those standards, and existing/extant metadata will require some acknowledgement of these tensions and judicious application of metadata cross-walks where conflict is unavoidable. **The recommendations herein seek to minimize such cross-walks and align as much as possible with DSpace default configurations.**

As an additional disclaimer, some of the namespaces and crosswalks utilized in DSpace feature a mix of camel-case, lower-case, or other conventions for recorded metadata. You will see *some* of this reflected in this document. When in doubt, doublecheck your metadata registry in DSpace and work with whatever is *default* in your repository. If adding a metadata element to items you intend to crosswalk to other schema, check the crosswalk for expected case, and/or contact someone at Scholaris support for confirmation.

About Dublin Core

Dublin Core is the metadata schema employed by the DSpace platform. It is managed by the Dublin Core Metadata Initiative (DCMI). In practice, Dublin Core is straightforward. It's a relatively small set of original *elements*, subdivided by *qualifiers* for specificity.

> **Simple Dublin Core** properties are those original 15 elements first defined by the Dublin Core Metadata Workshop Series in the mid-1990s. Though this categorization is conceptually useful, Simple Dublin Core is a somewhat deprecated notion now subsumed into the dc/terms/namespace as high-level properties.
>
> **Qualified Dublin Core** properties are refinements of the original 15 elements (with some elemental additions defined more recently by the DCMI), which are presently managed in the dc/terms/namespace.
>
> **Custom Dublin Core** properties are custom refinements of the controlled dc/terms/elements made by local schema developers. While the Dublin Core standard

> does not allow for custom elements to be asserted, it does allow for the custom refinement of existing elements through the Dumb-Down Principle. This principle states that local refinements on the Dublin Core elements are supported as long as external applications can “ignore any qualifier and use the description as if it were unqualified” [7]. Adherence to this principle ensures that all enterprise-specific metadata elements can be dumbed down and ingested by external systems, even with some loss of specificity, since all custom elements are sub-properties of controlled Dublin Core elements.
>
> From: https://doi.org/10.1002/bul2.2017.1720430211

The format for Dublin Core metadata is as follows:
`namespace.element.qualifier`

The namespace represents a metadata schema and its specifications. The element is the core metadata being recorded. The qualifier is for specificity. Because all Dublin Core formatted metadata is intended to accommodate the dumb-down principle, all qualified metadata *should reasonably pair with the parent element*. Additionally, anything without a qualifier is referred to as "unqualified Dublin Core."

For example, `dc.subject.lcsh` is an example of qualified Dublin Core: `lcsh` is the qualifier that is added to the existing namespace (`dc`) and element (`subject`). But, the metadata that would be described with `dc.subject.lcsh` would be equally appropriate metadata for a `dc.subject` element without the qualifier. A qualifier should always narrow down an element. Additionally, a custom *qualifier* is allowed in Dublin Core. Custom *elements* are not.

Recommended Minimum Metadata Elements and Scope Notes

All the tables within this section are viewable with additional details in an annex with additional language around mandatory, recommended, and optional metadata elements. The Minimum Viable Metadata Recommendations – ANNEX is available in the Scholaris Network of Experts Reports collection.

Generic Dublin Core for DSpace

Presuming that some institutions may defer from specific external indexing/harvesting needs, this is a recommended spec for a generic publication deposit leveraging Dublin Core *only*. In some cases, these recommendations might look like a best-case scenario for the content being migrated into the Scholaris service. However, not all repositories coming to DSpace will have been using Dublin Core prior. Some variations or additional elements may be necessary to keep useful metadata available.

Contributor Metadata

Dublin Core utilizes a single string for contributor metadata, be it the name of a person or an organization. It is typically written in `lastname, firstname` format. Users migrating from a more granular schema (such as MODS) may need to concatenate their existing metadata into this format.

Whereas "roles" or "affiliations" may exist as *attributes* of a specific author in other schema, in Dublin Core we are limited specifically to a single qualifier. For example:

- Author | `dc.contributor.author`
- Editor | `dc.contributor.editor`
- Translator | `dc.contributor.translator`

Date Metadata

DSpace has two kinds of date metadata out of the box. One kind is generated by the system automatically when content is ingested. The other is part of date metadata provided via submission forms. For example:

- `date.accessioned` | automatic based on ingest date
- `date.issued` | publication date

It is important to note that *any cross-walk* looking for unqualified Dublin Core at dc.date will attempt to pull all qualified values, and in some cases may pull only the earliest date (ex. LAC harvesting). Sometimes the *first* `dc.date` value is pulled instead. Metadata in DSpace is ordered alphabetically. It may be necessary to modify a cross-walk with a specific qualifier for disambiguation.

Rights (Access/Licenses)

Rights information is vital metadata for repositories. Access rights typically contain metadata related to the accessibility of an object to the public or otherwise. For example, whether the article is open access, embargoed, restricted, etc. Rights statements, on the other hand, are typically metadata related to *copyright* or license information about an item. That could be a creative commons license or other rights-related statement.

Much of the license metadata recorded would likely be relegated to an attribute of another element in a more granular schema. Or, such metadata might be obfuscated by the repository platform itself. One typical application is as follows:

- `dc.rights.uri` | URI for a Creative Commons License
- `dc.rights` | language of the CC license

However, many institutions have multiple, qualified metadata elements in this space related to any number of potential rights holders or various parties, all of which may seem *equivalent* to an organization pulling from unqualified `dc.rights` metadata. Because of the variety of interpretations in play across repositories in Canada, **the best recommendation here is to preserve internal consistency with a given repository**. Rights elements *will very likely require* XSLT Crosswalks for OAI-PMH to adequately push expected metadata downstream.

Related Works and Citation Metadata

Issue- or title-level metadata for repository items is typically, but inconsistently, stored in a few different ways. Again, what would be important here is internal consistency.

The Dublin Core element `dc.relation` is commonly used for information about a journal in which an article appears. For example, `dc.relation.ispartof` may be used to capture title-level metadata. Including a full citation for a journal article in a repository is also common within repositories, where it might be recorded in `dc.identifier.citation`, `dc.identifier.bibliographicCitation`, or `dc.relation.isversionof`.

Generic Dublin Core for Scholaris Repositories	
field	**scope notes**
MANDATORY	

Generic Dublin Core for Scholaris Repositories	
field	**scope notes**
dc.contributor.*	A person, organization, or service responsible for the content of the resource. Qualifiers for specific contributor types like, author, editor, translator, advisor… etc.
dc.creator	Utilized in OAI-PMH cross-walks by DSpace. Use dc.contributor.* qualified elements by default.
MANDATORY	
dc.date.accessioned	Date DSpace takes possession of the Item.
dc.date.issued	Date of publication or distribution.
dc.description.abstract	Abstract or summary, if available.
dc.format	Catch-all for any format information not defined by qualifiers (ex. dc.format.extent). This field is not without issues - formats not included in the registry are auto-populated as “unknown”. See also the LAC requirement for harvesting theses & dissertations for the recommendation to fix this dc.format crosswalk.
dc.language.iso	Current ISO standard for language of intellectual content, including country codes (e.g. "en_US"). S-MDEG recommends using a two-letter code from ISO 639-1 if available and appropriate, and if not, then using a more specific three-letter code from ISO 639-3.
dc.publisher	Entity responsible for publication, distribution, or imprint.
dc.title	Title statement/title proper.

Generic Dublin Core for Scholaris Repositories	
field	**scope notes**
dc.type	Nature or genre of content. Recommendation for adoption of COAR Resource Type CV.
RECOMMENDED	
dc.description.sponsorship	Information about sponsoring agencies, individuals, or contractual arrangements for the item.
dc.identifier.*	Additional identifiers as applicable, with qualifiers per ID type, including: doi, uri, url, govdoc, isbn, issn, ismn
dc.rights.*	Terms governing use and reproduction.
dc.rights.uri	License metadata.
dc.rights.holder	The owner of the copyright.
dc.subject.*	Relevant qualifiers might include: keywords, lcsh, csh, or other classification types.
OPTIONAL	
dc.relation.*	Related resource(s) Relevant qualifiers might include: isPartOf, hasVersion

There are use cases where a link can or should be made to another item in the repository, or at another location (e.g., a dataset for a thesis that is stored in a data repository). This may also be

captured using `dc.relation`, or stipulated with relationships via Configurable Entities. General guidance from the DSpace community is to record publisher/external DOIs in `dc.relation.hasversion` (this is the default behaviour as of D10).

OpenAIRE Compliant Spec

OpenAIRE is a European non-profit that exists to promote and facilitate open science (which includes open access). One of the ways they do this is by providing the OpenAIRE Guidelines, which are intended to help repository managers expose publications, datasets and CRIS metadata via the OAI-PMH protocol. This, in turn, facilitates integration with OpenAIRE infrastructure, which includes discovery layers. If you are unfamiliar with the Guidelines, we recommend you read at least the Introduction to understand the context for their use.

The OpenAIRE application profile includes the following namespaces:

dc: http://purl.org/dc/elements/1.1/
dcterms: http://purl.org/dc/terms/
datacite: http://datacite.org/schema/kernel-4
oaire: http://namespace.openaire.eu/schema/oaire/

Compliance with these OpenAIRE Guidelines will be desirable for many repositories moving to DSpace primarily because of the improved discovery options and interoperability with other infrastructure. As we will continue to say across all our recommendations documents, we believe it is best to adhere to existing guidelines and standards wherever possible to improve interoperability and to reduce the potential for error (in this case with custom OAI-PMH mapping). Additionally, one significant benefit to considering OpenAIRE compliance is that the `oaire` namespace allows for more granular publication metadata than has generally been possible with *just* Dublin Core. As a result, some metadata elements and their local use may differ. Some specific examples are outlined below.

These recommendations are based on evaluation of both the OpenAIRE v4.0.0. Application Profile Overview and the default XSL Crosswalks bundled with DSpace >7.0.

Contributors

As with generic Dublin Core, the implementation of OpenAIRE utilizes a single string for contributor metadata, be it the name of a person or an organization. It is typically written in `lastname, firstname` format. Users migrating from a more granular schema (such as MODS) may need to concatenate their existing metadata into this format.

However, for OpenAIRE harvesting to work, Configurable Entities will likely need to be enabled. In the DSpace 7 OpenAIRE Compliance documentation (the DSpace 8 version is the same as 7 at time of writing) it states that Entities are required for OpenAIRE 4 functions. This is because, when the Entities models and fields are correctly configured, a *repository* can then be configured to expose their metadata in the OpenAIRE format. This will include the virtual metadata (metadata that is dynamically determined at the time of access based on an Entity's relationship to other Entities). This virtual metadata can provide more granular and elaborate contextual and relational metadata. For example, Dublin Core is very bad for affiliation data related to authors of works. Virtual metadata can allow you to attach additional, relevant metadata to records. You can learn more about virtual metadata and Entities in the S-MDEG "Entities" Guidance and Advice document available in the Scholaris Network of Experts Reports collection or in the DSpace 8 Configurable Entities documentation.

Entities aren't a *requirement* of OpenAIRE per se, despite the fact that they will definitely make things easier. In theory, an institution could ensure that they are providing metadata that includes all of the OpenAIRE mandatory fields and relationships in the format the OpenAIRE Application Profile requires without Entities, but *we don't recommend this approach*. Users aiming to work with OpenAIRE that skip Entities will end up with significantly more customization in their deposit forms and, possibly, a temptation to stretch Dublin Core beyond its intended use.

DSpace >7 expects author metadata for default use to better facilitate downstream crosswalks.

- Author | `dc.contributor.author`
- Editor | `dc.contributor.editor`
- Translator | `dc.contributor.translator`

Rights (Licenses)

In DSpace 8, *access* rights are stored and can be disseminated by the platform via OAI without a need to record that information as Dublin Core metadata. However, license information still requires specific attention. Because of the addition of the OpenAIRE namespace in DSpace 7-8, users are afforded a wider range of potential rights options for metadata. The default interpretation is as follows:

- `oaire.license.condition` | URI for a Creative Commons License

However, an impending deprecation of the Creative Commons API, which is used for the long-standing deposit step known as the Creative Commons License Picker has some implications that complicate recommendations in this space. The picker has used `dc.rights` and `dc.rights.uri` for license information, in tandem. This behaviour has been hard-coded

into DSpace. If you want a CC license label on records in your repository, you'll still have to put license language in the `dc.rights` field and the CC license URI in the `dc.rights.uri` field. Either way, *check with Scholaris to see what the status of these changes is*. Upcoming versions of DSpace may well require alternative solutions and there is active discussion in the DSpace community to look at approaches to replicate or replace the current CC license chooser functionality.

As with a generic Dublin Core application in DSpace, there is no specific, universally accepted element for something like a "rights statement" from a publisher. For these, it's recommended to employ a custom qualifier. In the wild, currently, there are many different applications in this space. What is vital is their consistent use in a repository and their ability to adhere to the dumb down principle. Some approaches taken by Canadian repositories include:

- `bul.rights.addendum` | employing a custom namespace, Université Laval uses this field for rights information.
- `bul.rights.raisonEmbargoInfini` | employing a custom namespace, Université Laval uses this field for rights language on work that's indefinitely embargoed in their repository.
- `dc.description.copyright` | employing a custom qualifier, the University of New Brunswick uses this element for publisher copyright statements pertaining to accepted manuscripts and preprints.

Related Works and Citation Metadata

The OpenAIRE namespace included with DSpace 7 dramatically improves issue- and title-level metadata options for publications in a repository. These include:

- `oaire.citation.conferenceDate` | The date a conference took place.
- `oaire.citation.conferencePlace` | Where a conference took place.
- `oaire.citation.edition` | The edition the work was published in.
- `oaire.citation.endPage` | Page range metadata.
- `oaire.citation.issue` | Issue of the container.
- `oaire.citation.startPage` | Page range metadata.
- `oaire.citation.title` | Title of the container (book, journal, conference).
- `oaire.citation.volume` | Volume of the container.
- `oaire.fundingStream` | Funder metadata.
- `oaire.version` | Version metadata as per COAR controlled vocabulary.

In the context of Scholaris, repositories migrating from a more granular metadata standard may be able to port their metadata to these elements. Migrating such metadata from current Dublin Core among adopters on DSpace 5 or 6 may prove more challenging.

What follows is the required metadata for OpenAIRE. You'll notice a number of these entries are in the *datacite* namespace. OpenAIRE will leverage the Datacite cross-walk to write these values by default.

Metadata for OpenAIRE	
field	**scope notes (from OpenAIRE Documentation)**
MANDATORY	
datacite:creator	The authors of the publication in priority order. May be a corporate/institutional or personal name.
datacite:contributor*	The institution or person responsible for collecting, managing, distributing, or otherwise contributing to the development of the resource. Contributor qualifiers require terms approved by OpenAIRE or those within CRediT.
datacite:date (Embargo)	The second provided date in an OpenAIRE deposit will be assumed to be an embargo availability date.
datacite:date (Publication)	A date associated with an event in the life cycle of the resource. Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [W3CDTF] and follows the YYYY-MM-DD format.
MANDATORY	
datacite:identifier	Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Example formal identification systems include the

Metadata for OpenAIRE	
field	**scope notes (from OpenAIRE Documentation)**
	Uniform Resource Identifier (URI), the Uniform Resource Locator (URL), the Digital Object Identifier (DOI), and the URN:NBN. Also, this can be a direct URL, or a redirection URL, like PURL, HANDLE or other international resolution mechanisms.
datacite:rights (access)	In the OpenAIRE spec, there is a presumption here of "Access Rights" language as per COAR.
datacite:subject	
datacite:title	
dc.description	An account of the content of the resource. Description may include, but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.
dc.format	Provided systematically by DSpace. Should export according to MIME Type definitions.
dc.language	
dc.publisher	An entity responsible for making the resource available. Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.
MANDATORY	

Metadata for OpenAIRE	
field	**scope notes (from OpenAIRE Documentation)**
oaire.file	Provided systematically by DSpace.
oaire:fundingReference	Information about financial support (funding) for the resource being registered.
oaire.resourceType	Nature or genre of content. Recommendation for adoption of COAR Resource Type CV.
RECOMMENDED	
dc.identifier.oclc	OCLC Number.
dc.identifier.uri	Uniform Resource Identifier
dc.identifier.url	URL for version of record or if published elsewhere.
dc.language.iso	Current ISO standard for language of intellectual content, including country codes (e.g. "en_US").
dc.relation.hasversion	The DOI of the version of record for a publication.
datacite:alternateIdentifier	Per DataCite: An identifier other than the primary Identifier applied to the resource being registered. May be used for local identifiers, a serial number of an instrument or an inventory number, and should be an additional identifier for the same instance of the resource (i.e., same location, same file).
OPTIONAL	
dc.subject.*	
OPTIONAL	

Metadata for OpenAIRE	
field	**scope notes (from OpenAIRE Documentation)**
oaire.citation.conferenceDate	The date when the conference took place. This property is considered to be part of the bibliographic citation. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [W3CDTF] and follows the YYYY-MM-DD format.
oaire.citation.conferencePlace	The place where the conference took place. This property is considered to be part of the bibliographic citation.
oaire.citation.edition	The edition the work was published in (e.g. book edition). This property is considered to be part of the bibliographic citation.
oaire.citation.endPage	The end page is part of the pagination information of the work published in a container (e.g. journal issue). This property is considered to be part of the bibliographic citation.
oaire.citation.issue	The issue of the container (e.g. journal). This property is considered to be part of the bibliographic citation.
oaire.citation.startPage	The start page is part of the pagination information of the work published in a container (e.g. journal issue). This property is considered to be part of the bibliographic citation.
oaire.citation.title	The title name of the container (e.g. journal, book, conference) this work is published in. This property is considered to be part of the bibliographic citation.
oaire.citation.volume	The volume, typically a number, of the container (e.g. journal). This property is considered to be part of the bibliographic citation.
oaire.license.condition	Creative Commons and other license dropdown elements as required by OpenAIRE.

Metadata for OpenAIRE	
field	**scope notes (from OpenAIRE Documentation)**
OPTIONAL	
oaire.version	Use either a version number or the label of the vocabulary term as value. For example, COAR version type vocabulary based on NISO Journal Article Version.

Theses & Dissertations / LAC Spec

For many institutions, theses & dissertations collections make up a substantial portion of repository collections. And, because theses & dissertations come with no small quantity of institutional baggage, unique processes, political challenges, and localized solutions to any of the above, consistent metadata across institutions can be something of a challenge.

S-MDEG, with explicit feedback from the Scholaris Electronic Theses & Dissertations Expert Group (S-ETEG), has made recommendations for Thesis/Dissertation metadata to align as maximally as possible with the ETD-MS cross-walk and the thesis harvesting requirements of Library & Archives Canada. For the list of fields and their uses, please consult the LAC requirements for harvesting theses & dissertations available in the S-ETEG Scholaris Network of Experts Reports collection. We have also captured the list of fields in the Minimum Viable Metadata Recommendations – ANNEX (available in the S-MDEG Scholaris Network of Experts Reports collection) in the lac tab, which provides the most current scope notes and mapping.

On Forms

For many repository managers, **forms** are the main interaction they have with DSpace metadata. In DSpace, forms are largely determined by two files:

`dspace/config/item-submission.xml` | The file that stipulates forms by item type, and contains all the rules for display steps therein.

`dspace/config/submission-forms.xml` | The file that contains all of the direct mappings to metadata schema of form elements. This one file will contain form information for *all potential*

forms you use, as well as any controlled vocabulary “value pairs” for drop-down fields you might configure locally or otherwise.

Your ability to modify these files will depend on the circumstances of your repository's installation. With hosted services like Scholaris, you can ask for configuration changes and they have ample advice. On the other end of the spectrum, your institutions may host DSpace locally and give you direct access to these XML files. What you use for forms is *enormously dependent* on your workflow at your institution. Even within the S-MDEG group, we found a wide range of approaches. Some examples include:

- Simple submission forms for logged-in users who deposit their own works, where metadata is adjusted further in the workflow by library or other associated staff/users.
- Platforms with wholly external deposit forms where DSpace forms become less-seen by users and contain all relevant metadata, filled out by repository managers or staff.
- Platforms that leverage DSpace workflows for a hybrid of these two models.
- Platforms that primarily pull metadata from other systems and crosswalk to Dublin Core.
- Any number of changes to `submission-forms.xml` as they might relate to locally-relevant metadata and – in particular – custom metadata related to ETDs.

Note: The S-ETEG has published approaches and considerations for ETD Workflows on TSpace.

The forms you choose to use will stem from the metadata decisions you make. **The Scholaris team can provide repository teams with template forms based on the recommendations in this document.** They also have submission form guidance in their own documentation. Their choices include:

- **<Recommended if using Entities> OpenAIRE Publication form** | This form features more granular publication metadata than the traditional DSpace forms, specifically better journal- or title-level metadata for publications. Though it presumes a connection to the OpenAIRE crosswalk, it also leverages the more specific and accommodating `oaire` namespace allowing for better overall metadata quality.
- **Publication - Standard Form** | A form meant to leverage publication metadata from the Publication Entity type. Has more granular metadata than a traditional DSpace publication form, but is less capable of specificity than the OpenAIRE form.
- **Traditional Submission Form** | Meets all stock Dublin Core metadata recommendations of this document and is the most basic of the three forms. Minimum viable metadata.

It's vital to note that all three of these forms presuppose a lack of custom metadata and are unlikely to require any modification to DSpace crosswalks.

Threading the Needle: Cross-walks and Implementation of OpenAIRE/ETD-MS/Rioxx or Other Schema

Generally – outside the historic strata of metadata practices within a given institution over time – repositories store metadata in *one schema* and can export it to others. The process by which metadata is shuffled from one schema to another is called a **cross-walk**. Cross-walking metadata is a process of taking specific metadata elements and mapping their contents to comparable or equivalent metadata elements in another schema.

Here's an example of cross-walked metadata. On the left, we have MODS metadata for an author of a work. On the right, we have author metadata in Dublin Core.

<mods:name type="personal"> <mods:role> <mods:roleTerm authority="author" type="text"> <mods:namePart type="given">Suzie</mods:namepart> <mods:namePart type="family">Smith</mods:namepart> </mods:roleTerm> </mods:role> </mods:name>	<dc:contributor.author>Smith, Suzie</dc:contributor.author>

In this example, the metadata on the left is more granular than the metadata on the right. Metadata stored as an XML schema attribute in MODS is replaced with Qualified Dublin Core. And, because Dublin Core name elements are a single string, the name is concatenated as `last, first` as part of the move.

A Word of Caution

You can change virtually any of these mappings in DSpace by modifying an XSLT (Extensible Stylesheet Language) file. This is great if you are very certain of the changes you are making. But, ***you can also thoroughly ruin things if you are not careful***. When working with DSpace

configuration files, you can *always* go back to DSpace defaults by pulling those files from GitHub. But, going back to your own modified config files *will require you to be diligent with backups and cautious with changes*.

Broadly speaking, you may not need to touch these files at all. Or, you may enlist the help of the Scholaris team when doing so.

In DSpace, crosswalks are configured with **XSLT files** within the config directory. Stock crosswalks for the platform can be seen at the following link: https://github.com/DSpace/DSpace/tree/main/dspace/config/crosswalks

XSLT is a language for transforming XML documents into other formats (e.g., valid XML transformed to valid HTML) or into other XML documents (e.g., a valid MODS XML document transformed to a valid MARC XML document). This is done by specifying the source metadata (elements, attributes, etc.) and defining what that metadata should look like in its destination format. For example, here is a snippet of the DSpace default ETD-MS XSLT file:

```
<xsl:for-each
select="doc:metadata/doc:element[@name='dc']/doc:element[@name='contributor']/doc:
element[@name='author']/doc:element/doc:field[@name='value']">
<creator><xsl:value-of select="." /></creator>
</xsl:for-each>
```

This snippet indicates that the content of each instance of `dc.contributor.author` should be output within a `dc.creator` element.

There are many good resources out there for learning about and working with XSLT, such as the W3C XML and XSLT tutorials. And many colleagues within the Canadian repository community who can lend a hand!

It's likely most users won't change these XSLT files much themselves, but they are modifiable. In the primary directory, you can see some examples of metadata destinations pre-configured against the default DSpace form language.

- ./crosswalks/DIM2DataCite.xsl | Maps default Dublin Core to DataCite's Schema for registering DOI metadata.
- ./crosswalks/DIM2MARC21slim.xsl | Dublin Core to MARC21.

- ./crosswalks/oai/metadataFormats/oai_openaire.xsl | The crosswalk for DSpace's OpenAIRE OAI endpoint.
- ./crosswalks/oai/metadataFormats/etdms.xsl | The crosswalk for the ETD Metadata schema that's leveraged, in part, by Library & Archives Canada for thesis harvest.
- ./crosswalks/oai/transformers/rioxx.xsl | The cross-walk for RIOXX (Research Outputs Metadata Schema, a schema that looks to bridge the gaps *between* schemas like Dublin Core and MODs) mapping.

By default, these cross-walks all operate under a presumption of *default form and metadata applications within a repository*. But, clearly, many DSpace repository managers make changes to DSpace's metadata registry. Or, over time, specific qualified Dublin Core ends up subsumed by alternative namespaces, varying qualifiers, or completely custom metadata to meet more local repository needs.

Regardless, it's vital to note that **changes to the application of Dublin Core metadata elements within a repository will *also* require reviewing existing crosswalks for desired downstream metadata use**.

DSpace 7/8 has a practical example of implementation drift that this experts group has seen at tension. Most interpretations of Dublin Core would store author metadata in the element `dc.creator`. But, out of the box, DSpace 7/8 forms put author metadata in the qualified element `dc.contributor.author`. Furthermore, default crosswalks for OpenAIRE, Google Scholar, ORCiD, ETD-MS and others *all* expect author metadata in `dc.contributor.author`. **A change to the way authors are recorded in DSpace 7/8 has clear implications for all metadata cross-walked out of the platform**.

Some repositories use `dc.creator` for primary author metadata. This isn't inappropriate, but it would likely be useful for migrating institutions to pivot author metadata to `dc.contributor.author` to avoid having to change every crosswalk in the system to accommodate. DSpace expects author metadata in this qualified element and has already preconfigured cross-walks accordingly.

Controlled Vocabularies

Rights (Access/Licenses)

It is likely that many migrants to DSpace may not have this information as *metadata*, per se, but more likely its information stored in *their respective platforms* that may or may not be exportable.

For example, Islandora 7.x stores access rights information at the bitstream level, not in metadata. DSpace 7 stored this as metadata and, as of DSpace 8, can now store it as system metadata instead. That is, the OAI cross-walk for OpenAIRE can interpret resource policies (including embargo conditions) in DSpace and record the corresponding access rights downstream (DSpace 8 Release Notes).

Complicating things a little further, DSpace itself is often configured with a License Chooser form step leveraging the Creative Commons API, where `dc.rights` and `dc.rights.uri` are populated automatically. The License Chooser feature will be deprecated by the end of 2026, which only raises further headaches.

This tension makes any recommendation outside of the OpenAIRE interpretation a bit of a moving target. Ultimately, what's most vital is that you have rights, a url for a license, and that these elements are being:

- recorded consistently across all items in your repository,
- cross-walked responsibility where possible/relevant,
- documented internally in your metadata registry (e.g., in a Scope Note), and
- machine readable (stick to controlled vocabularies where possible and URL/URI).

Disciplines / Subjects

In a discovery layer situation, a shared Controlled Vocabulary for subjects would be super ideal. We might be able to cull some inspiration from OpenAlex's *Topics*.

Language (dc.language.iso)

DSpace defaults to two-letter language codes in the ISO 639-1 specification. As outlined in the S-MDEG Multilingual Repositories Recommendations available in the Scholaris Network of Experts Reports collection (echoing COAR Multilingual Repositories recommendations), the S-MDEG recommends using a two-letter code from ISO 639-1 if available and appropriate, and if not, then using a more specific three-letter code from ISO 639-3. The working group acknowledges that, although ISO 639-3 is more in-depth regarding dialects, micro- and macro-languages, and other specifics, it remains *deficient* for describing many of the indigenous languages used by First Nations communities across North America.

Obviously, it would be a kind of madness to recommend adoption of the full set as controlled vocabulary, but there should be a general adoption of the most commonly used languages per repository.

The code tables for the subset can be found at this link:
https://iso639-3.sil.org/code_tables/639/data

Downstream ingest of ISO 639 language codes may be largely dependent on cross-walks. OpenAIRE, for example, utilizes the RFC 4646 spec to identify language metadata.

Funder Metadata

Forthcoming changes to the Canadian federal research Tri-Agency OA policy, coupled with the work and advocacy of the Barcelona Declaration, has put some emphasis on the importance of good metadata for funded research. Broadly speaking, the best-case scenario for funder metadata would include:

- **Grant DOI** | A DOI registered against an entire grant. The Tri-Agency only *just* started with Grant DOIs and are following in the steps of the FRQ (Fonds de recherche du Québec) here. This is an emerging use case, but one supported by major PID registration agencies.
- **Grant IDs** | Grant IDs are more commonly used than Grant DOIs. They're funder-dependent and often have no associated metadata you can pull. In situations where a researcher doesn't have a Grant DOI, they may well have a Grant ID.
- **Funder ID** | Broadly speaking these are best recorded as a ROR (Research Organization Registry) ID, but in some cases could be plain text, or a Ringgold ID, or leverage the deprecated Crossref Funder Registry. In any event, this ID field is meant to identify the funding body.

In the current recommended mapping, we've flagged `oaire.fundingReference` which looks at `dc.relation`. The dumb-down principle could support any number of elements here in a relation field, but not in a way that's especially parsable downstream.

Because of the flat nature of Dublin Core metadata, and the likelihood of publications with multiple sources of funding, the best potential application at the moment would probably require the application of Organization and Project Entities in order to facilitate better virtual metadata applications. They can leverage ROR values, and more specifically tie funding bodies to Projects and potentially Publications. There may be additional implications for this metadata as DSpace merges with DSpace CRIS over the next few releases of the software.

At best, a solution for repositories that aren't using Entities is to place the name of the funder or the ROR of a funder in the `dc.relation` element as stipulated in the minimum viable documentation.

Type (dc.type)

The working group recommends adoption of the COAR Resource Type Vocabulary. Or, at minimum, we recommend a subset of the resource type vocabulary.

Version Types

In the case of OpenAIRE integration, the working group recommends adoption of the COAR Version Type Vocabulary for journal articles or other peer-reviewed content. Based on NISO Journal Article Version recommendations, this metadata is expected in `oaire:version` metadata. It's worth noting that NISO JAV is currently under revision, and is *really* meant to be used for *only article versions*.

Ambitions Outside Scope

The following includes minimal recommendations that may fall out of scope, but S-MDEG members have expressed a desire to flag some more aspirational metadata goals should significant changes of DSpace appear on the horizon.

Date Format

The working group recommends expressing date formats with an international standard to ensure clarity, maintain consistency within the repository, and to interoperate with other systems. The committee suggests using either ISO 8601 or EDTF depending on local context as these are commonly used standards. OpenAIRE suggests using the DCMI definition and usage instructions for date, where the "recommended best practice for encoding the date value is defined in a profile of ISO 8601 [W3CDTF] and follows the YYYY-MM-DD format." Further, they note regarding "fuzzy" time periods, "to express more about that temporal period, one can use the `dc:coverage` field."

Dates can refer to many different periods in the life cycle of a resource (ex. publication date, availability date, etc.). OpenAIRE guidelines suggest the following: use DataCite elements for embargo dates (`datacite:date`), for property dates use the publication date as value. For attribute type dates, choose terms from the date type vocabulary, using the `"Issued"` value for publication date. OpenAIRE has three terms in its date type vocabulary: Accepted, Available, and Issued.

Example:

```
<datacite:date dateType="Issued">2000-12-25</datacite:date>
```

Note OpenAIRE date types differ slightly from DC qualifiers dc.date.accessioned, dc.date.available dc.date.issued and attention might need to be paid during any cross-walking.

DSpace is consistently developed with any necessarily time cross-walks in mind, *so long as you don't modify the default settings*.

The working group is not yet sure of capability here with default or otherwise unmodified DSpace deployments. Committee members do note, however, complications with EDTF. While EDTF provides a great deal of flexibility, it can pose challenges when defining the level of precision of dates required in the submission process, or in determining sort-by-date order. While these are not insurmountable challenges, they should be considered and balanced against your particular use cases.

Collections Hierarchy

Many institutions have taken the opportunity presented by a migration or major version upgrade to clean up or simplify collection hierarchies. After a generation of repositories sorted by faculty and departmental hierarchical collections/communities, there's a pivot away from "browse by" priorities to more streamlined structures with fewer Communities/Collections (e.g., organized by resource type instead of by department or faculty, which can change and require more maintenance over time). If navigating by department or faculty is still desirable, additional search filters can be added to provide this functionality. Some schools taking such an approach thus far include:

- Université Laval
- University of New Brunswick
- University of Guelph
- University of British Columbia (v. 5)

Recommendations on collections are, we acknowledge, politically fraught. However, a looming migration *is an excellent time to consider a change to the hierarchy while you're moving an entire repository's worth of content*.

On Responsible Custom Metadata

There are many reasons for a repository to accommodate custom metadata. In terms of migrations, some metadata schemas may be more granular than Dublin Core. Take MODS for

example. MODS is based on MARC 21 and has an extensive tag set related to an enormous volume of MARC fields. A "note" element in MODS might not have a clear-cut Dublin Core equivalent, but someone moving to Dublin Core might also not want to jettison any metadata.

Metadata for Theses and Dissertations is another space where any number of custom metadata fields might be recorded for local record keeping or posterity. Information on graduation dates or faculty/departmental information is often relegated to custom metadata and collections hierarchies.

Though you may not need as much custom metadata as you're inclined to record, there are two rules of thumb for custom metadata:

1. If the metadata you need to record will be cross-walked out of your repository to other systems, and it's *close enough* to standard Dublin Core that its use fits the "dumb-down principle", you can use a custom *qualifier*.

2. If the metadata you need to record is *only used locally* and is necessarily specific, you may want to employ a custom *namespace*.

Custom Qualifiers

A custom qualifier can alleviate the need for creating a custom namespace for metadata in your repository, while also aligning more cleanly with the existing Dublin Core schema. Metadata needs *will not* be universal across all institutions. Some repository managers may need to record metadata that doesn't really have an obvious matching metadata element.

A custom qualifier allows a repository manager to make a specific piece of metadata *more specific*. This might be useful for search faceting or navigation. It might also be useful for recording metadata that Dublin Core may not afford its own specific element to.

But, it's vital to remember that *any* custom qualifier *must fit within the scope of the element it is qualifying*.

Custom Namespaces

If qualified Dublin Core is found to be inadequate for your metadata model, you have the option of using properties from existing namespaces. This way of customizing your model complies with the Linked Open Data (LOD) best practice of reuse. Whenever possible, opt for LOD vocabularies that have a wide adoption, as well as evidence of regular maintenance in order to assure the longevity of your choices for your model.

There may be cases where you need to capture metadata about a resource and find that there is no appropriate property in existing LOD vocabularies. In this case, you can create and use custom namespaces. While there are benefits to custom namespaces, there are also risks, and so before doing so it is good practice to really assess the need for the metadata in question, and to do a thorough examination of existing vocabularies. It is equally important to follow good community practice (see for example the W3C Best Practices for URI Construction or the European Data Portal Design & Manage Persistent URIs), and to document your decisions. Out of box, DSpace includes a 'local' namespace, however you can also set up a custom namespace with an institutional identifier to make it more evident that the metadata is institution-specific/sourced (example 3 below).

Examples

Example 1. Custom Qualifier Replacing Faculty/Department

During UNB's migration from Islandora to DSpace, they decided to get away from custom faculty and department metadata. Often, these sorts of metadata were met with requests, clarifications, and other issues. The hierarchical nature of the varying university structures was a lot more fluid than one might expect. So, in the move, they made the call to pair a custom qualifier with university stipulated graduate academic units (GAU).

UNB's GAUs were typically named after the lowest tier of a Faculty/Department Hierarchy. For example:

- Faculty of Engineering
    - Department of Mechanical Engineering (GAU = Mechanical Engineering)
    - Department of Chemical Engineering (GAU = Chemical Engineering)
- Faculty of Computer Science (GAU = Computer Science)
- Faculty of Arts
    - Department of History (GAU = History)
    - Department of Classics (GAU = Classics)

Subject metadata is often pretty varied and is a mix of both controlled and uncontrolled vocabulary. Keywords aren't usually controlled. Subject headings aren't "controlled" but *are* significantly more intentional. UNB decided to use a custom qualifier in the subject field and mapped it to all GAUs at their institution. For this, they use `dc.subject.discipline` (ex. Full Record page).

This works as a "by subject" filter for the whole repository, instead of a loose set of uncontrolled terms. It also was convenient in mapping the etdms/thesis namespace for the LAC-required `thesis.degree.discipline` element.

The qualifier itself is custom metadata, and its use in UNB's repository is quite specific, but because it is a repeatable element and one that dumbs down cleanly to `dc.subject`, it's not problematic as a custom metadata solution.

Example 2. Custom Qualifier to Accommodate Migration from MODS.

When UNB migrated from Islandora to DSpace, it required a migration from MODS to Dublin Core. Because MODS is more granular *and* based on MARC 21, some descriptive metadata did not have a clear location. Two notable examples follow:

- `dc.description.note` | MODS utilizes a MARC notes field typically leveraged by cataloguers for local, contextual information about a record. These didn't totally fit in `dc.description` by themselves, but also weren't inappropriate for the element in general, so a custom qualifier of `.note` was applied for these.

- `dc.description.copyright` | The Copyright Office at UNB Libraries felt it would be useful to specifically store license statements from publishers in a unique, specific metadata element

Example 3. Custom Property to Capture Thesis/Dissertation Specialization

At the University of Alberta, there was a need to be able to capture specialization information, which is something that only some programs had. (For example, this thesis is a Doctor of Philosophy in Animal Science from the Department of Agriculture, Food, and Nutritional Science). Their current repository was custom-built in-house and, though they tried to leverage LOD vocabularies whenever possible, they couldn't find a really suitable property at the time. Instead, they created a custom property in their local namespace (UAL:specialization) and for OAI harvest mapped it to https://www.ndltd.org/standards/metadata/etd-ms-v1.00.html (with prefix Specialization in content).

Example 4. Using Existing LOD Vocabularies to Capture Degree, Department, and Faculty

When McGill Libraries migrated from Digitool to Hyrax in 2019-2020, several custom properties from the original metadata profile (which was mainly in unqualified Dublin Core, with the custom properties named as localthesisdegree etc.) were mapped to existing linked data vocabularies. For example, the properties for Faculty, Department, Degree, Research Unit, Grant Number, and Local Affiliated Centre were mapped to the following existing vocabularies:

Property label	Property

Faculty	frapo:Faculty
Department	frapo:Department
Degree	bibo:degree
Research unit	frapo:ResearchGroup
Grant number	frapo:hasGrantNumber
Local affiliated centre	vivo:Center