

Metadata Draft Copy

Migration to GeoNetwork 3



Funded by

Environment Canada



Author: GeoCat bv

Last Updated: 2/12/2015

Table of Contents

[1 Overview \(for proposal\)](#)

[2 Description of the Functionality](#)

[2.1 Metadata Draft Copy](#)

[2.2 Extended Workflow](#)

[3 Current Implementation in 2.10.x](#)

[4 Migration Process](#)

[4.1 Edit](#)

[4.2 Store Draft](#)

[4.3 Search and view](#)

[4.4 Publish](#)

[4.5 Design](#)

[4.5.1 EditWorkspace Interface](#)

[4.5.1.1 getQueryBuilder](#)

[4.5.1.2 getStatusActionsFactory](#)

[4.5.1.3 onSave](#)

[4.5.1.4 onEdit](#)

[4.5.1.5 onDelete](#)

[4.5.2 IndexUtils Interface](#)

1 Overview (for proposal)

This proposal enhances the metadata workflow in GeoNetwork to provide a draft copy of metadata for editing and a published copy once the metadata is set to published. This allows to make changes to editors in the metadata without affecting the published copy.

This work is based on the original work done for AGIV (Belgium) and updated/improved later for a version based on 2.10 for Environment Canada.

Thanks also to Jesse Eichar for suggestions about refactoring the feature for GeoNetwork 3.0.

The following sections describe in detail the technical implementation.

2 Description of the Functionality

This functionality adds two features to the core 2.10.x version of GeoNetwork:

2.1 Metadata Draft Copy

When a published metadata is edited, a new copy is saved on both the database and the index marked with the draft flag. This way a user with enough privileges can view, search and edit published metadata without modifying what the rest of the users see about this metadata.

This draft is blocked while a user is editing it to prevent other concurrent editing with other users. This block is released either when the editor is closed or after a timeout of inactivity.

Once the draft copy is published, the original published metadata gets replaced with the draft copy and the draft is removed from the system.

2.2 Extended Workflow

The basic workflow on GeoNetwork is extended so it has now two levels of publication: internal and external.

3 Current Implementation in 2.10.x

Based on commit 793321d033cabef4242ad8db5ce6f57f1d149aa3.

Files involved:

- web/src/main/java/org/fao/geonet/GeonetContext.java
- web/src/main/java/org/fao/geonet/constants/Edit.java
- web/src/main/java/org/fao/geonet/constants/Params.java
- web/src/main/java/org/fao/geonet/kernel/AccessManager.java
- web/src/main/java/org/fao/geonet/kernel/DataManager.java
- web/src/main/java/org/fao/geonet/kernel/MdInfo.java
- web/src/main/java/org/fao/geonet/kernel/XmlSerializer.java

- `web/src/main/java/org/fao/geonet/kernel/XmlSerializerDb.java`
- `web/src/main/java/org/fao/geonet/kernel/XmlSerializerSvn.java`
- `web/src/main/java/org/fao/geonet/kernel/csw/services/Transaction.java`
- `web/src/main/java/org/fao/geonet/kernel/csw/services/getrecords/CatalogSearcher.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/arcSde/ArcSDEHarvester.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/csw/Aligner.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/fragment/FragmentHarvester.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/geonet/Aligner.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/geonet20/Aligner.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/localfilesystem/LocalFilesystemHarvester.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/oaipmh/Harvester.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/ogcwxS/Harvester.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/thredds/Harvester.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/webdav/Harvester.java`
- `web/src/main/java/org/fao/geonet/kernel/harvest/harvester/z3950/Harvester.java`
- `web/src/main/java/org/fao/geonet/kernel/mef/Importer.java`
- `web/src/main/java/org/fao/geonet/kernel/search/LuceneIndexField.java`
- `web/src/main/java/org/fao/geonet/kernel/search/LuceneIndexWriterFactory.java`
- `web/src/main/java/org/fao/geonet/kernel/search/LuceneQueryBuilder.java`
- `web/src/main/java/org/fao/geonet/kernel/search/LuceneSearcher.java`
- `web/src/main/java/org/fao/geonet/kernel/search/SearchManager.java`
- `web/src/main/java/org/fao/geonet/kernel/search/UserQueryInput.java`
- `web/src/main/java/org/fao/geonet/services/config/Set.java`
- `web/src/main/java/org/fao/geonet/services/metadata/AjaxEditUtils.java`
- `web/src/main/java/org/fao/geonet/services/metadata/BatchOpsMetadataReindexer.java`
- `web/src/main/java/org/fao/geonet/services/metadata/DefaultStatusActions.java`
- `web/src/main/java/org/fao/geonet/services/metadata/Delete.java`
- `web/src/main/java/org/fao/geonet/services/metadata/EditUtils.java`
- `web/src/main/java/org/fao/geonet/services/metadata/GetEditableData.java`
- `web/src/main/java/org/fao/geonet/services/metadata/GetLocked.java`
- `web/src/main/java/org/fao/geonet/services/metadata/GrabLock.java`
- `web/src/main/java/org/fao/geonet/services/metadata/Insert.java`
- `web/src/main/java/org/fao/geonet/services/metadata/Show.java`
- `web/src/main/java/org/fao/geonet/services/metadata/StatusActions.java`
- `web/src/main/java/org/fao/geonet/services/metadata/StatusActionsFactory.java`
- `web/src/main/java/org/fao/geonet/services/metadata/Unlock.java`
- `web/src/main/java/org/fao/geonet/services/metadata/Update.java`
- `web/src/main/java/org/fao/geonet/services/metadata/UpdateAdminOper.java`

- `web/src/main/java/org/fao/geonet/services/metadata/UpdateCategories.java`
- `web/src/main/java/org/fao/geonet/services/metadata/UpdateStatus.java`
- `web/src/main/java/org/fao/geonet/services/metadata/XslProcessing.java`
- `web/src/main/java/org/fao/geonet/services/ownership/Transfer.java`
- `web/src/main/java/org/fao/geonet/services/util/ServiceMetadataReindexer.java`
- `web/src/main/webapp/WEB-INF/classes/setup/sql/create/create-db-default.sql`
- `web/src/main/webapp/WEB-INF/classes/setup/sql/create/create-db-postgres.sql`
- `web/src/main/webapp/WEB-INF/classes/setup/sql/data/data-db-default.sql`
- `web/src/main/webapp/WEB-INF/config-lucene.xml`
- `web/src/main/webapp/WEB-INF/config-metadata.xml`
- `web/src/main/webapp/WEB-INF/config-security-mapping.xml`
- `web/src/main/webapp/loc/eng/xml/config.xml`
- `web/src/main/webapp/loc/fre/xml/config.xml`
- `web/src/main/webapp/scripts/config/model.js`
- `web/src/main/webapp/scripts/config/view.js`
- `web/src/main/webapp/xsl/config/config.xsl`
- `web/src/main/webapp/xsl/metadata-grab-lock-results.xsl`
- `web/src/main/webapp/xsl/metadata-grab-lock.xsl`
- `web/src/main/webapp/xsl/metadata-status.xsl`
- `web/src/main/webapp/xsl/metadata-utils.xsl`
- `web/src/main/webapp/xsl/rest/md-context.xsl`
- `web/src/main/webapp/xsl/rest/search-results.xsl`

4 Migration Process

The draft functionality for metadata can be added to the core GeoNetwork functionality, where a simple configuration may enable or disable this feature.

The extended workflow functionality can be also migrated to version 3 of GeoNetwork, but it is too customized for Environment Canada to add it to the main GeoNetwork source code.

The code between version 2 and 3 has changed a lot, so a direct migration is not possible. On the other hand, this allows us to refactor and improve the initial design for the solution.

4.1 Edit

When the edit action is called, we have to check if the metadata is already published or not. If it is published, we will work over a draft copy. If there is no current draft copy, a new one will be created.

This change will affect the following functionality:

- Lock Editing: New procedure to run when editing. Assign edit privileges only to one user.
- Unlock Editing: New procedure to run after closing editor or timeout. Removes locks
- Check Lock: New procedure to check if a metadata is currently locked.

- Create new Draft: New procedure to create a new draft based on a published metadata.
- Open Editor: Choose between draft or unpublished metadata (if it is not locked). Call create new draft if needed. Lock it.
- Save: Choose between draft or unpublished metadata.
- Close Editor: Call unlock.

4.2 Store Draft

The draft will be stored on a separated table on the database. This table will have the same columns as the original metadata table. On the index, the draft will be saved using the same fields and configuration as the metadata, but with a flag marking it as a draft. This way, we can filter on search if we are looking at a draft or a metadata.

This change will affect the following functionality:

- Save metadata: add the draft flag and select the right table on the database
- Remove metadata: remove all drafts associated to the metadata

4.3 Search and view

When searching metadata, the system will check if the user has privileges to look for drafts on certain groups. This check can result on allowing the metadata indexed with the draft flag.

This change will affect the following functionality:

- Check Privileges: new procedure to check if the user can see drafted versions
- Search: Add (or not) draft flag to search
- View metadata: Disallow draft view based on “Check Privileges” functionality

4.4 Publish

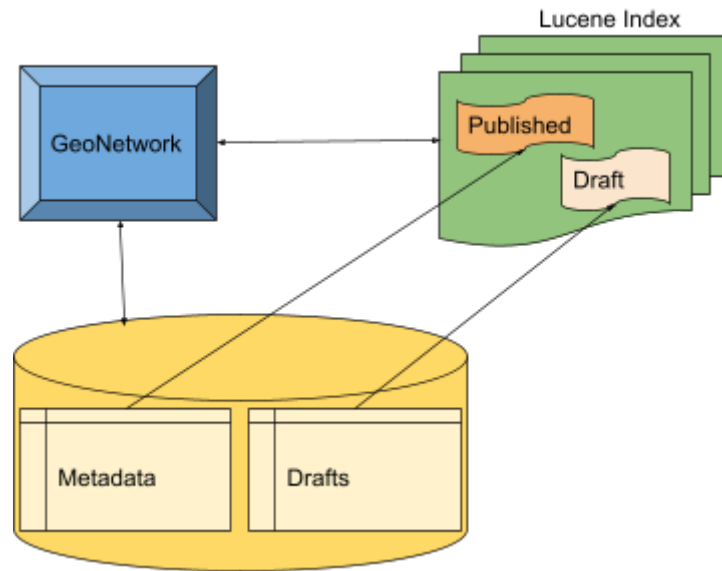
Once the draft copy is published, the original published metadata gets replaced with the draft copy and the draft is removed from the system.

This change will affect the following functionality:

- Publish: Replace metadata with draft (if exists) and remove draft

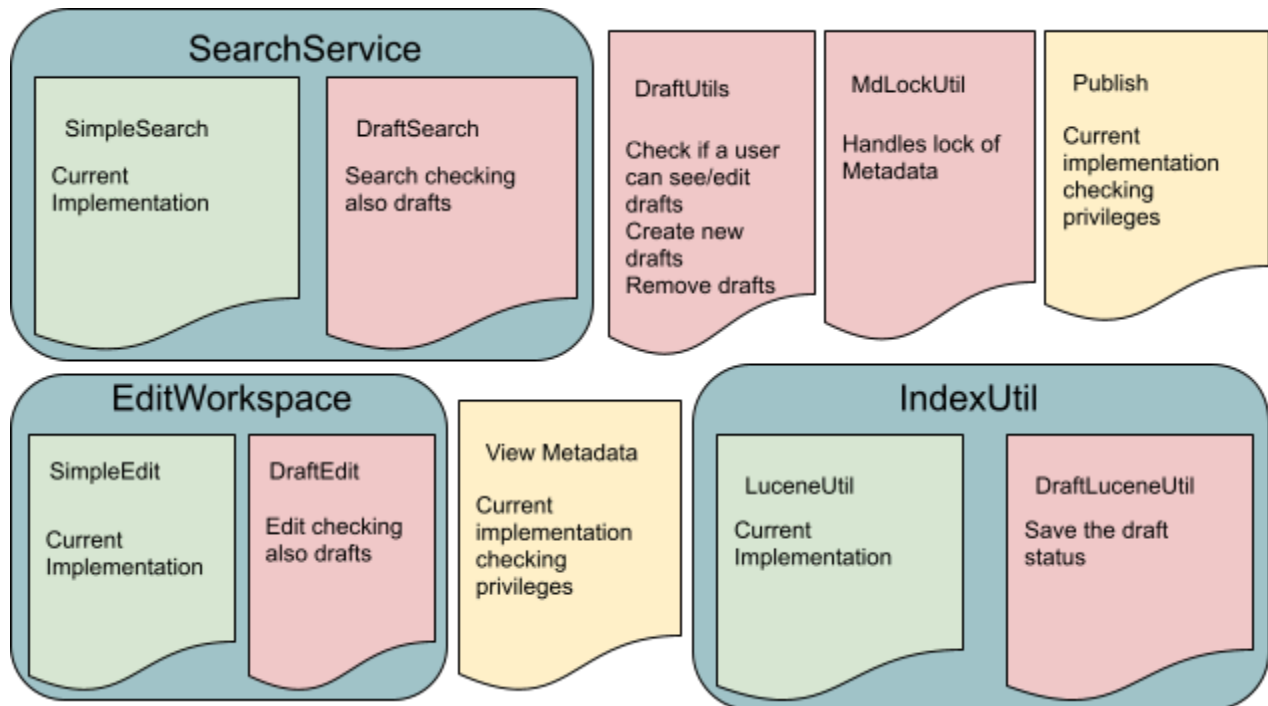
4.5 Design

A new table will be added on the database for the drafted copies. The Lucene index will be the same, except for a new boolean field that will indicate if the metadata stored is a draft or not.



There will be two different sets of classes on GeoNetwork: one that works with Metadata and another that works with drafts. If the draft extension is enabled on the configuration, GeoNetwork will have to load the set of classes that handles drafts.

The following image shows a simplified schema of the modifications needed to implement it. Not all classes are represented here, for example, both the LuceneSearcher, LuceneQueryBuilder and other Lucene related classes are represented by a single object called LuceneUtil. The different sets depending on the configuration are separated by blue rounded squares:



Events will also added on all the actions and status changes on metadata. This way, it will be easy to hook up later other kind of improvements to the metadata workflow, like improving the current SVN repository that contains the full history for the metadata.

4.5.1 EditWorkspace Interface

This interface will be an utility class for all metadata changes, both drafts and original metadata data. There will be a different implementation depending if the drafts are enabled or not.

This interface will be used for example on `org.fao.geonet.services.metadata.Update` and `Delete` classes so the services don't have to care if the metadata is a draft or not. It just relies on this utility class to get the beans or call the functions that takes care of the differences.

4.5.1.1 getQueryBuilder

LuceneQueryBuilder **getQueryBuilder**(Set<String> tokenizedFieldSet, Map<String, LuceneConfig.LuceneConfigNumericField> numericFieldSet, PerFieldAnalyzerWrapper analyzer, String langCode);

Returns the lucene query builder. It will use the current implementation if drafts are disabled and it will use the draft flag if the drafts are enabled.

@param tokenizedFieldSet names of tokenized fields

@param numericFieldSet names of numeric fields

@param analyzer Lucene analyzer

@param langCode language of search terms

4.5.1.2 getStatusActionsFactory

StatusActionsFactory **getStatusActionsFactory**();

Returns action factory to use for this workspace. The status allowed will depend if the drafts are enabled or not.

4.5.1.3 onSave

void **onSave**(boolean finished);

Called when save is called.

@param finished if true the editing is finished

4.5.1.4 onEdit

Metadata **onEdit**(int metadataId);

Called when editing has started.

@param metadataId Metadata that is going to be edited

4.5.1.5 onDelete

void **onDelete**(int metadataId);

Called when a metadata is deleted. It will take care of any draft copies available, if the draft is enabled.

@param metadataId Metadata that is going to be removed

3.5.1.5 onCancelEditSession

void **onCancelEditSession**(int metadataId);

Called when a metadata editing is cancelled. It will take care of any draft copies available, if the draft is enabled.

@param metadataId Metadata that was being edited

4.5.2 IndexUtils Interface

As with the previous described interface, this interface will be an utility class for all index operations, both drafts and original metadata data. There will be a different implementation depending if the drafts are enabled or not.

This interface will mostly replace the implementation for org.fao.geonet.kernel.search.LuceneQueryBuilder and provide utility functions for org.fao.geonet.kernel.search.SearchManager.