Advanced Embargo Support

The University of Michigan and @mire are initiating work to create a more robust and configurable embargo feature than the one that currently exists in DSpace. We would like to invite others to tweak the feature set if there's additional features of interest.

https://jira.duraspace.org/browse/DS-895

Business Requirements

Without going into detail about the technical implementation, the basic feature set we are looking at, which will build on the current implementation includes the ability to provide

1. Full embargo: no access to either the metadata or bitstream of an item until embargo period ends. Even if a match existsItems do not appear in: search results, browse results, RSS feeds, OAI-PMH feeds, etc. Though the metadata and bitstreams are stored, it's as if the item and its bitstreams had never been deposited, or indexed.
2. Partial embargo: metadata visible, some/all bitstreams inaccessible. In this case, items appear in browse or search indexes so searchers and browsers know an item and its bitstreams exist. However, they can't download/view bitstreams. (This is the behavior of the current implementation.)
3. Group access to embargoed items: embargoed items (full or partial) are viewable and all contents are completely accessible to members of select groups. This will address needs often expressed by archivists, who need a dark archive of materials in DSpace

So, when an item is under embargo:

- Whether the metadata page and all metadata fields are still visible to the world (and through OAI-PMH) should be configurable at the Item level.
- Whatever is viewable by an anonymous searcher and browser becomes part of the general search and browse indexes. Everything else is not.
- None of an item's bitstreams, including full text indexed by the system, are readable to the world under either a full or a partial embargo. Collection Administrators or pre-defined groups can bypass that protection.
- Any new bitstreams added to an item inherits that item's embargo settings.
- Administrators can change embargo status: either add an embargo to a currently open item, or release an item early.
- The presence of an embargo and its expiration date is visible in the administrative UI view of an item. Whether to display this to an end-user/non-administrator searcher or browser could be an option.

That's the basic idea, and I hope it's enough to give you a sense of where we're headed. The new embargo solution will be implemented for both XMLUI and JSPUI.

Again, we welcome comments on this and will do our best to incorporate your suggestions.

Business requirements for item restrictions in the LCoNZ repositories

The four repositories in the LCoNZ IRR project (AUT University's Scholarly Commons, Otago University's OUR Archive, Unitec Institute of Technology's Research Bank, Waikato University's Research Commons) have been using extensive customisations to item restriction settings for over a year. The business requirements for restrictions differ slightly across these institutions and have evolved over time. The following is an attempt to summarise these requirements. The requirements in the bulleted list above all apply as well.

The following item states are currently possible:

1. Open: The item metadata and bitstreams are fully visible and accessible. The item is discoverable through search and browse, is exported via OAI-PMH etc.
2. Abstract-only: The item metadata is fully accessible. Any bitstreams are visible to and accessible for repository staff only. The item is discoverable through search (but only the metadata is indexed) and browse, is exported via OAI-PMH etc.
3. Dark: The item metadata and bitstreams are visible to and accessible for repository staff only. The item can be found via browse, but only by repository staff. The item is not discoverable through search, browse or any other means by the general public and is not exported via OAI-PMH etc.

All three states are permanent in the sense that state changes are done manually whenever the need arises (eg at an author's request).

In addition to any of these three states, items can be either fully or partially embargoed. Embargoes have a set expiry date; however, embargoes never lift automatically. Instead, a notification is sent to a specific repository staff group whenever an item embargo is about to expire; repository staff typically liaise with other university staff to ensure that the embargo can actually be lifted (eg in the case of theses, where embargoes may be extended in some cases). The embargo applies until it is manually lifted by repository staff, even after the expiry date. Fully embargoed items are treated like dark items until the embargo is lifted. Partially embargoed items are treated like abstract-only items until the embargo is lifted. When the embargo is lifted, the item state changes to Open, Abstract-only or Dark as appropriate (though typical combinations are Full embargo -> Open, Full embargo -> Abstract-only, Partial embargo -> Open). Changes to the underlying permanent state of an item take the embargo status into account -- eg when a dark item is made open while it is under a full embargo, in practise it will not become open until the embargo is lifted (but then it will become open rather than dark).

In all cases, only those bitstreams from given bundles are publicly visible; others (such as preservation copies or administrative evidence files) are accessible to repository staff only. Whenever a new bitstream is added to an item, the item's state determines the access settings for that bitstream. Standard DSpace metadata visibility restrictions as set up in dspace.cfg apply to Open and Abstract-only items.

Introduction of per-bitstream embargoes (where some bitstreams are publicly accessible and others are suppressed) would be of interest to some of the LCoNZ repositories. Currently, the only distinctions made are repository (admin) staff vs general public, but extension of these principles to allow special access to specific groups (such as on-campus users or the institution's staff and students) might be of interest in the future.

The strict requirement that embargoes must require manual action stems from the fact that embargoes so far are used only for theses. There are plans to apply partial embargoes (the most recent addition to the LCoNZ restrictions model) to journal articles, to comply with publisher mandates. In this case, automatic embargo lifts on expiry may be desirable.

To summarise, in the LCoNZ model, items can either be open or restricted. Restrictions can be partial or complete; they can also be permanent ("until further notice") or temporary (with a fixed expiry date). Temporary restrictions still require manual action to be lifted. Any of the permanent item states can apply after an embargo is lifted, but not all combinations of partial and temporary restrictions are used in practice.

LCoNZ adds "Email based Embargo Termination Notification" and "Require Manual Embargo Release" business requirements to those already defined in general requirements. Another LCoNZ addition is the distinction of permanent state vs temporary restrictions (where the expiry of a temporary restriction changes the item permissions to the underlying permanent state).

Technical Requirements

Associated JIRA Tasks

DS-908 Embargo Overhaul: Utilize ResourcePolicy Start and Stop datestamps for enforcing embargo in DSpace

This approach is currently implemented in IDEALS and is based on existing Tapir embargo work that was completed there.

The adoption of this approach would allow for Embargo to be applied at either the Item level or individual Bitstream levels as a series of ResourcePolicies that use start/stop datestamps currently on the ResourcePolicy object.

Benefits:

- Does not require executing a cronjob to adjust the state of the Item
- Enforcement of embargo is stateless driven off the current date and the defined embargo time period.
- Resource Policies and Item State do not change over time
- Allows a Resource Policy to be a first class DSpace Domain Model citizen and that would include expanding it to have having name and description fields available to define the reason for a resource policy being set.

Establishment of a RESTRICT Action that would be enforced by the AuthorizationManager to allow for "Explicit Definition" of the Embargo Policy on Annonymous Users.

Example Use Cases:

Example 1 (Individual Bitstream Restriction)

Resource	Resource Policy Description	Resource Policies
Item A	Item itself is not embargoed	Action=READ, Group= Anonymous, start=null, end=null, name="Anonymous Read", description="Viewing Item is open to all Anonymous users."
Bitstream A.1	Restricted Access by Public Required by Publisher.	Action=RESTRICT, Group=Annonymous, start=20110101, end=20120101, name="Embargo", description="Restricted Access by Public Required by Publisher."
	Future Anonymous Access after Embargo Date	Action=READ, Group= Anonymous, start=20120101, end=null, name="Anonymous Read", description="Access Policies.pdf is available to all Anonymous users."
	Local University Affiliates are Exempt from Embargo Restriction.	Action=READ, Group=UniversityAffiliates, start=null, end=null, name="Local University Affiliates", description="Local University Affiliates are Exempt from Embargo Restriction."
Bistream A.2	Not Embargoed and Accessible to Public	Action=READ, Group= Anonymous, start=null, end=null, name="Anonymous Read", description="Access Policies.pdf is available to all Anonymous users."

Example 2 (Item Access Restriction)

Resource	Resource Policy Description	Resource Policies
Item A	Restricted Access by Public Required by Publisher.	Action=RESTRICT, Group=Annonymous, start=20110101, end=20120101, name="Embargo", description="Restricted Access by Public Required by Publisher."
	Local University Affiliates are Exempt from Embargo Restriction.	Action=READ, Group=UniversityAffiliates, start=null, end=null, name="Local University Affiliates", description="Local University Affiliates are Exempt from Embargo Restriction."
Bitstream A.1	Restricted Access by Public Required by Publisher.	Action=RESTRICT, Group=Annonymous, start=20110101, end=20120101, name="Embargo", description="Restricted Access by Public Required by Publisher."
	Local University Affiliates are Exempt from Embargo Restriction.	Action=READ, Group=UniversityAffiliates, start=null, end=null, name="Local University Affiliates", description="Local University Affiliates are Exempt from Embargo Restriction."
Bistream A.2	Restricted Access by Public Required by Publisher.	Action=RESTRICT, Group=Annonymous, start=20110101, end=20120101, name="Embargo", description="Restricted Access by Public Required by Publisher."

	Local University Affiliates are Exempt from Embargo Restriction.	Action=READ, Group=UniversityAffiliates, start=null, end=null, name="Local University Affiliates", description="Local University Affiliates are Exempt from Embargo Restriction."

The previous example would enforce Embargo and Access rights "Explicitly" and "Clearly" in the Policies attached to the Bitstream and/or Item. The AuthorizationManager may need minor enhancement to address "inheritance" of ResourcePolicies assigned on parent Items. It may be advisable to use such inheritance to enforce "DEFAULT_XXX" policies rather than copying them into place on each and every Bitstream/Bundle and Item created, this will reduce the "bloat" of ResourcePolicies currently in effect in the existing system.

And important benefit of these changes to ResourcePolicies and the underlying AuthorizationManager framework are that they can then be used to encode the explicit technical or administrative metadata sections into the AIP or METS manifests concerning the Policies that are in effect on the Item and its contents. Adjustments to the DSpace SIP Profile to capture enforcement of embargo details by consumers of those tools would be more clearly expressed and machine automatable than dumping it intot he metadata. Achieving Machine actionability means that Ingest Packagers and services that rely on them can define a more concrete business logic to be maintained.

As we evolve the Metadata capabilities to support system/tech/admin/descriptive metadata sections for all parts of the item, we can consider that the ResourcePolices will inform the production of metadata about the embargo state of the Item being exposed in OAI / SWORD /METS packagers and so-on. But for now, we really need to set a standard that actual Resource Policies be the mechanism that enforces the access rules/policies within the system and not some metadata field set in the item metadata description.

Somewhat a concern is how other areas of DSpace treat ResourcePolicies rather bluntly. Recommend that ResourcePolicies should be managed in central manner (such as ResourcePolicyService: or "ResorcePolicyManager") such that the manner in which policies are enforced or allowed to be edited does NOT cause emergent conflicting behavior across different parts of the system such as those described within DS-906 and DS-525.

According the DS-525, the issue of embargoed items is documented as a warning in our Documentation: https://wiki.duraspace.org/display/DSDOC/System+Administration#SystemAdministration-Movingitems

I consider this documentation insufficient as a solution to the problem of embargo permissions getting overridden in the mapping. A more appropriate solution would show to the user the exact changes that would happen to the item and allow them to decide which policies should be enforce/changed on the item.

Restricting Discovery of Embargoed Items in Search and Browse

Tapir/IDEALS Solution:

Item will need to be restricted from search and browse, the Tapir solution for restricting Item from viewing in Search and Browse is to set inArchive=false on the Item and add it to a "restricted" table for access in the "Embargoed Item" Administrators View. An "EmbargoLifter" command is responsible for removing these embargo settings from the Item to lift the embargo

General Solution:

A possibility in the general approach will be not to add the item to a restricted table,, but continue to encode the embargo period and conditions in the ResourcePolicies and create a view over the Items based on this criteria (ideally by using Discovery to indicate that either the Item or one of its Bitstreams is embargoed)

Restricting viewing of Embargoed Items

Resource Policies are currently set into place on DSpace Bitstreams during the desired period defined in the (usually **dc.date.embargountil**

Problem: Lifting the embargo removes any record of the original embargo being in effect/

Solution: Use of the provided "Start" and "End dates on the Resource Policies would allow the creation of an Embargo Period that could be preserved with the Item and would not need physical alteration to release the Item from Embargo.

DSpace Items Private

The AuthorizationManager already supports the enforcement of timeframes in ResourcePolicies. I would like to propose that we expand ResourcePolicy in the following manner:

Encoding of Embargo Rights in AIP

It should be possible to export and restore Embargo Rights on Bitstreams and Items based on PREMIS RIghts sections in the AIP

https://wiki.duraspace.org/display/DSDOC18/DSpace+AIP+Format#DSpaceAIPFormat-ExampleofMETSRightsSchemaforanItem

Migration from existing Embargo solutions

Metadata Concerning the setting of an embargo term and lift date may still be recoreded in the metadata. ALterations may include:

1. Not relying on date described in metadata to manage access rights
2. Exposure of the embargo details for any item in OAI/METS, XMLUI/METS or JSPUI Item views by evaluation of the ResourcePolices that are in force.

Related Projects and Solutions

IDEALS Embargo Functionality

The current //DSpace 1.5.2 //version has been customized with an embargo feature that allows three different group restrictions during the submission process:

Limit access to a group: this sends an email to an administrator, so the administrator can set this up manually
University Only access: either IP based or login
Complete embargo: the item is fully withdrawn until the embargo is released.

The embargo lift date is currently not defined as a date, but stored as a number of months. It is stored as a date in the database after selecting the amount of months. The submitter also needs to supply a reason for the embargo. In the item metadata it should be exposed that the item is under embargo, and also mention the embargo lift date.

There is also a custom mechanism for batch uploads, and a script for mapping the correct embargo settings on these uploaded items, and the import mapfile. Embargo will also effect OAI-PMH exposure of content through withdrawn/archived item state. Also the submitter should be able to access the item. The client requests these functionalities to be retained.

At the moment, only entire items can be embargoed, but the client is optionally interested in being able to set an embargo on the bitstream level as well. Optionally the client is requesting an implementation to support for current user to contact submitter for “request to access” the embargoed content.

Description

The current implementation of the embargo functionality can be migrated to DSpace 1.6.2, but requires some optimizations to limit changes to the DSpace core classes.
@mire recommends the client to invest in re-working the embargo implementation to decrease any further upgrade issues to future versions of DSpace, which can be achieved using two different approaches. The approach @mire advises, is detailed here, and the second approach is described in Alternative 1.

The advised solution does not use the embargo framework because there are limitations to the extensibility of this framework. We will create a custom solution which limits changes to existing DSpace core classes as much as possible, and maintains the same storage solution for the embargo settings from the current repository. Both the interface during the submission and the edit interface (for admin & submitter) for archived items will be ported.

Custom Embargo Per Bitstream

Embargo per bitstream instead of per item. This solution will be an extension to the solution (and alternative) mentioned above. The main difference is that all embargo functionality will need to be entered and stored per uploaded bitstream (file). Both the interface during the submission and the edit interface for archived items will be adjusted to ensure the settings can be configured for each bitstream individually.

Embargo per Item - Technical description

The current solution provides a form in which the submitter can indicate if the item is private or pubic. If private is chosen at least one of the following private options has to be selected:

- Visible to University of Illinois users ONLY
- Visible to a Smaller Group of users ONLY
- Make publicly visible after a period of time (field to indicate number of months)

In case the user selects one of the first two options ("//Visible to University of Illinois users ONLY//" or "//Visible to a Smaller Group of users ONLY//") the following operations are performed:

- a record is added into *__restrict_item__* table
- a metadata field is added *dc.provenance.description* (*Item marked as restricted to the 'UIUC Users.....*)

The Item results to be:

- SEARCHABLE
- WITH FILES RESTRICTED

In case the user select only the third option ("//Make publicly visible after a period of time//") the following operations are performed:

- a record is added into *__restrict_item__* table
- a record is added into *__bi_embargoed__* table
- a metadata field is added *dc.provenance.description* (*Item marked as restricted to the 'UIUC Users.....*)

The Item results to be:

- NOT SEARCHABLE

Summarizing the previous description, we have:

Option	=record in //restrict_item//	=record in //bi_embargoed//	=Searchable ?	= Bitstream(s) Visible?
Visible to University of Illinois users ONLY	Yes	No	Yes	No

Visible to a Smaller Group of users ONLY	Yes	No	Yes	No
Make publicly visible after a period of time (field to indicate number of months)	Yes	Yes	No	-

The tables involved in the current solution have to following structure:

RESTRICT_ITEM

=Field	=Description
id	unique identifier
item_id	unique item identifier
release_date	filled only if "//Make publicly visible after a period of time//" is chosen
eperson_group_id	unique group identifier
reason	reason why the access at the item is restricted

BI_EMBARGOED

=Field	=Description
id	unique table identifier
item_id	unique item identifier
sort_1	item title
sort_2	item date issued
sort_3	item data available

Embargo per Bitstream - Technical description

The new solution provides the users with two ways to define restricted access at bitstream level:

- Restrict Access
- Upload/Edit File

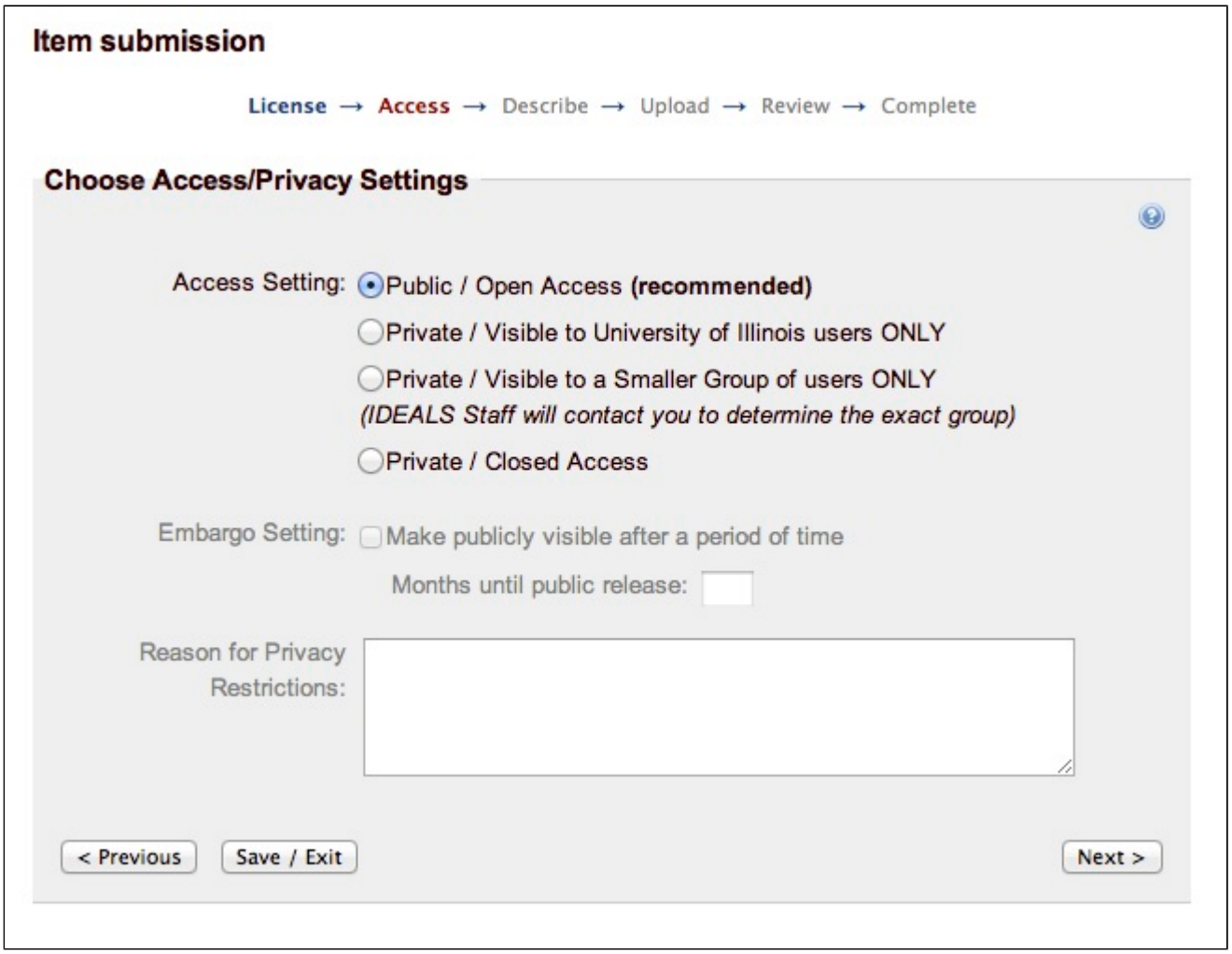
These functions can be activated in:

- UI Submission
- Edit Item
- Workflow

Restrict Access

This function is the same contemplated in the previous version, but the form has been modified.

At this level the user can define general access settings options. This means that every new bitstream uploaded by default will have the same access settings defined by "Restrict Access" function.

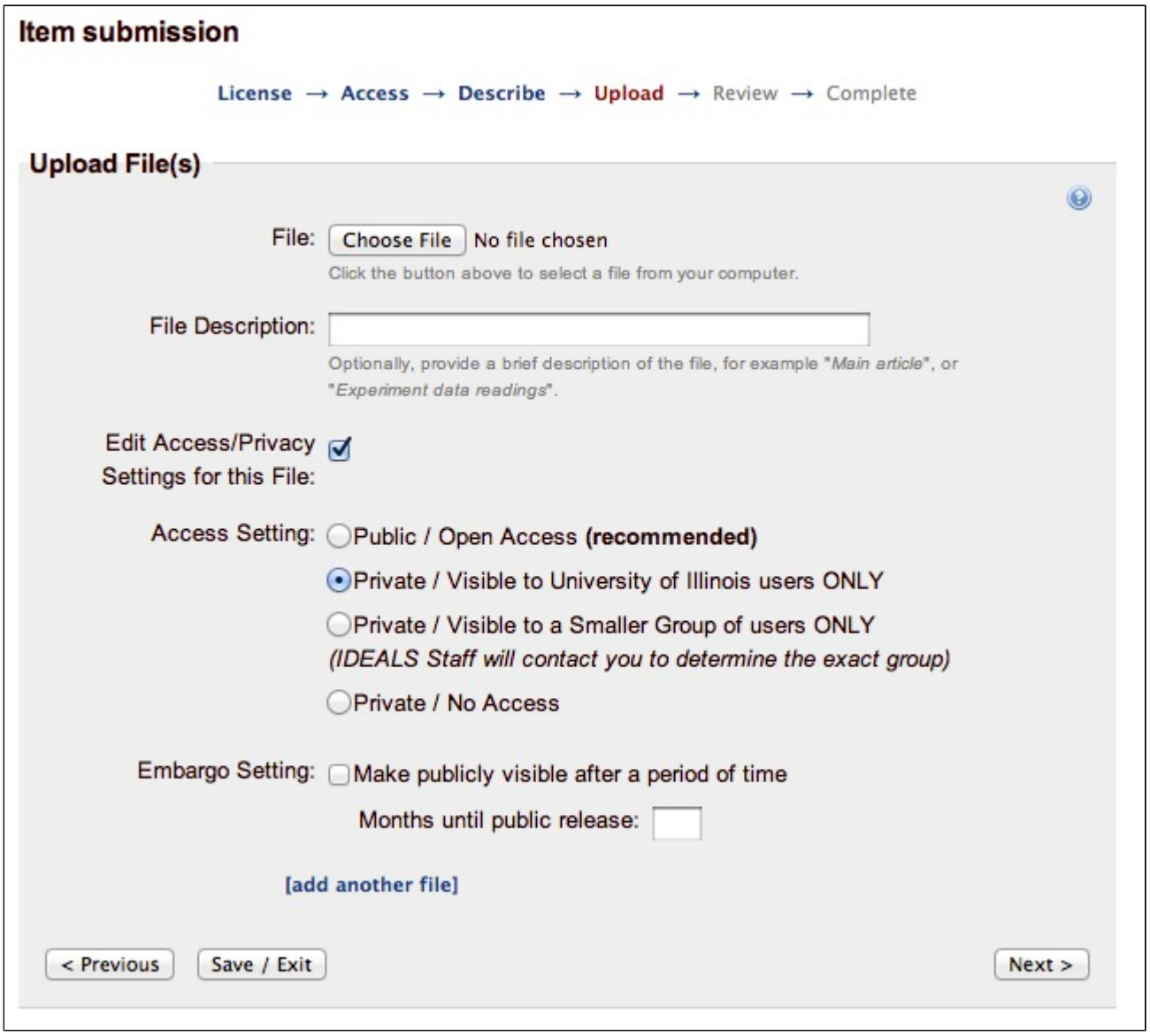


__Note:__

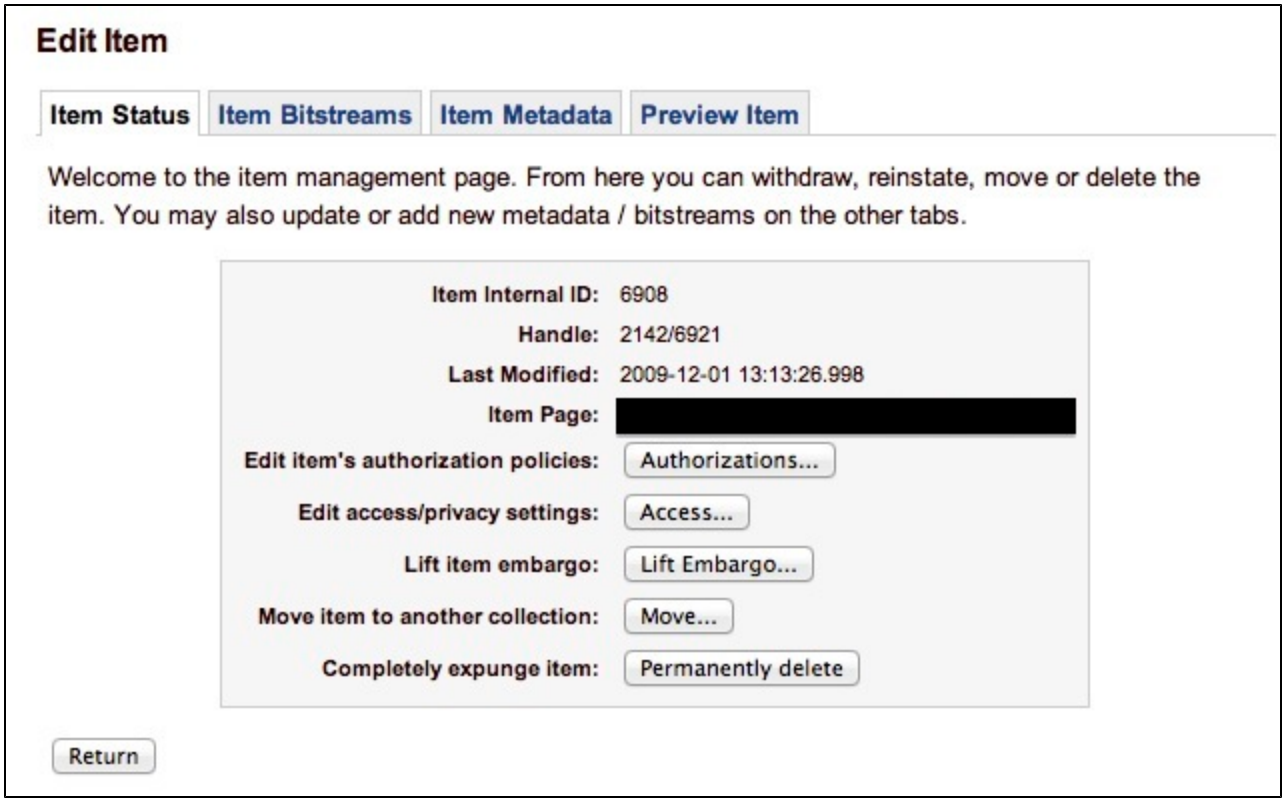
1. *__If the user will change the general access settings during the workflow or after the item is installed the new settings will be applied to all the bitstream(s) attached to the item.__*
1. *__If the general settings are "//Private/Closed Access"// won't be possible define custom settings at bitstream level.__*

Upload/Edit File

In the modules //UploadStep// and //EditItem// the user can define the custom access settings per bitstream, overriding the general defined previously.



Administrative Item Access



To override the general settings the user must to check "//Enable Custom Settings//".

Back-End

The settings are defined in the table //resourcepolicy, //here a summary about what happens when the settings are defined:

= SETTING	= ACTION in RESOURCEPOLICY
PUBLIC	A record with DEFALT_READ Anonymous policy is added
PRIVATE_NO_EMBARGO	A record with DEFALT_READ Group policy is added
PRIVATE_EMBARGO	A record with DEFALT_READ Group is added A record with START_DATE is added

Note: The class //InstallItem// has been overridden because when the Item came installed the policies of all the bitstreams came overridden.

Comments:

Current Workflow Help Documentation

https://services.ideals.illinois.edu/wiki/bin/view/IDEALS/IDEALSHelpPage#Submit_Choose_Access_and_Privacy

Feature supports placing item into withdrawn.

Embargo by Item only applies to Items that are Open Access, Visible to a smaller group or university of illimois users. This is because in those cases the item is in an archived state in the repository.

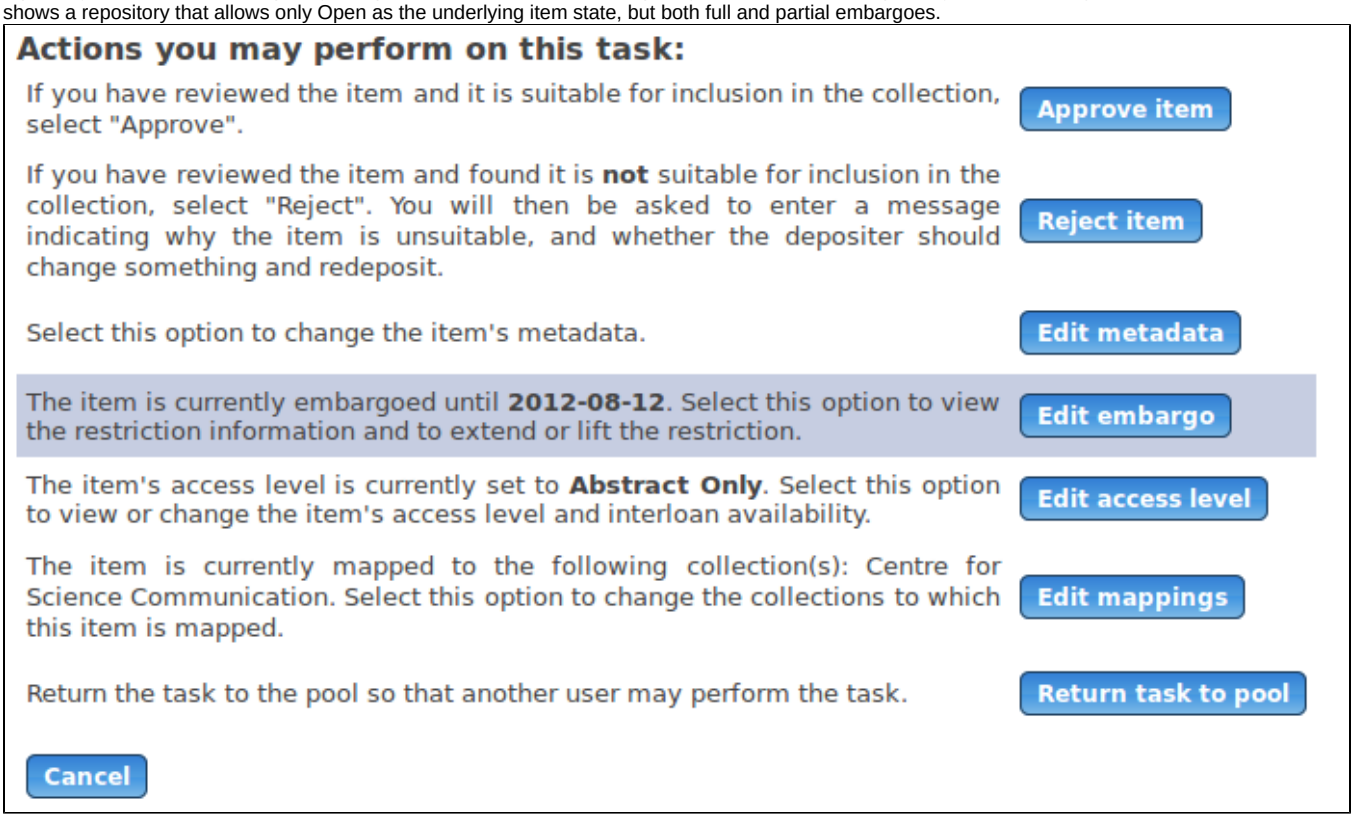
Item restrictions in the LCoNZ IRRs

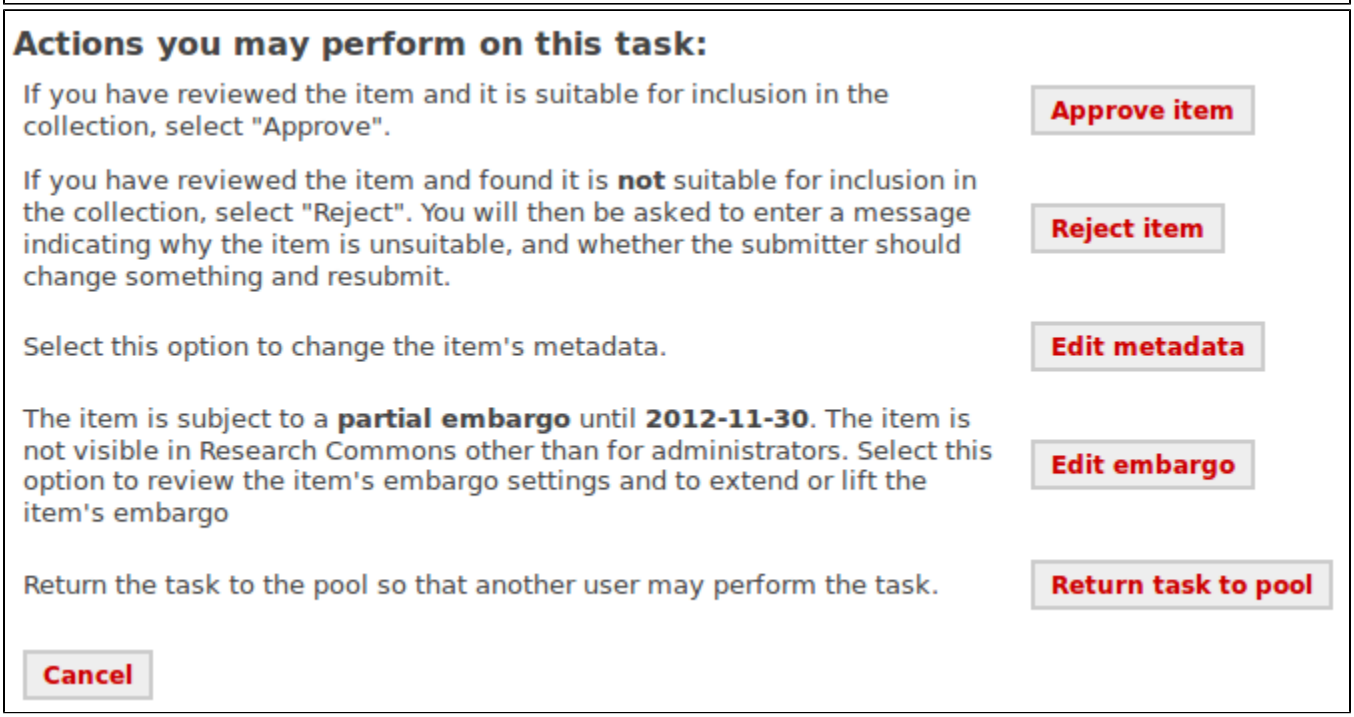
See Business requirements for item restrictions in the LCoNZ repositories above. The description in this section is for DSpace 1.7.2 and XMLUI.

User interface description

The restriction type as well as the embargo type + expiry date can be set in the DSpace deposit form as well as the SWORD thesis deposit forms, with the exception of dark items. Items deposited into a certain community in one of the repositories (which happens via a third-party application) are automatically turned dark.

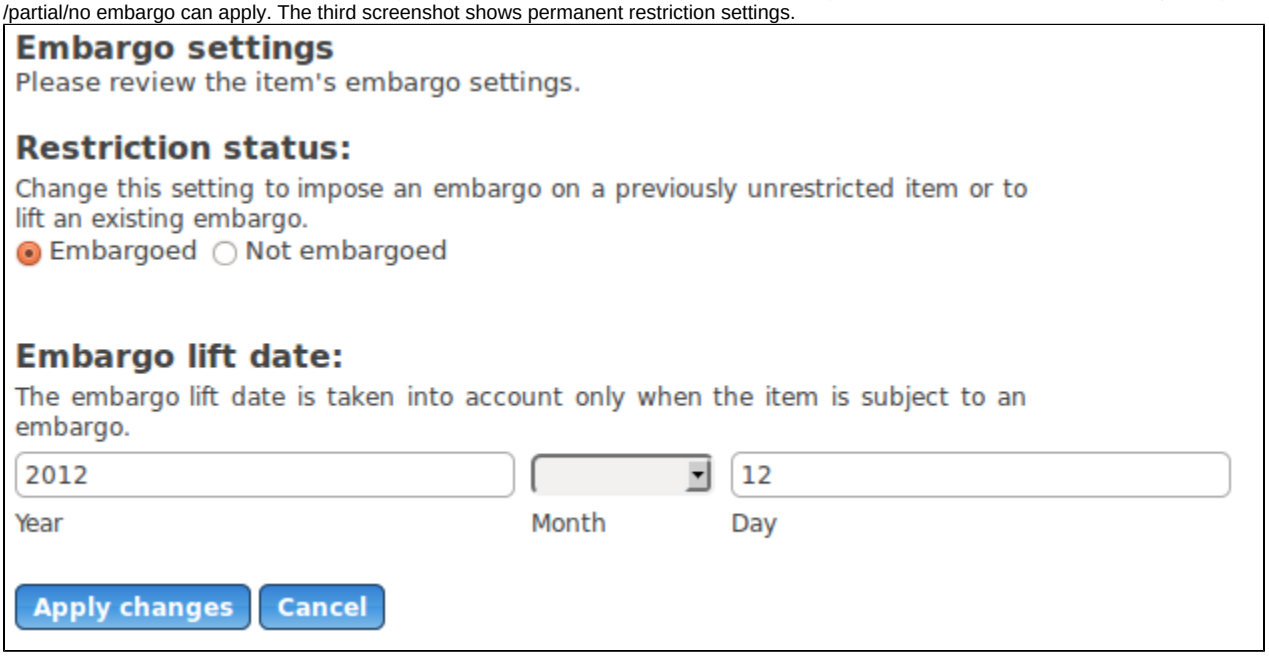
Repository staff can review and change an item's restriction status in the workflow or via "edit item" once the item is archived. The first screenshot shows a repository that allows both temporary and permanent restrictions; all temporary restrictions in this repository are full embargoes. The second screenshot shows a repository that allows only Open as the underlying item state, but both full and partial embargoes.

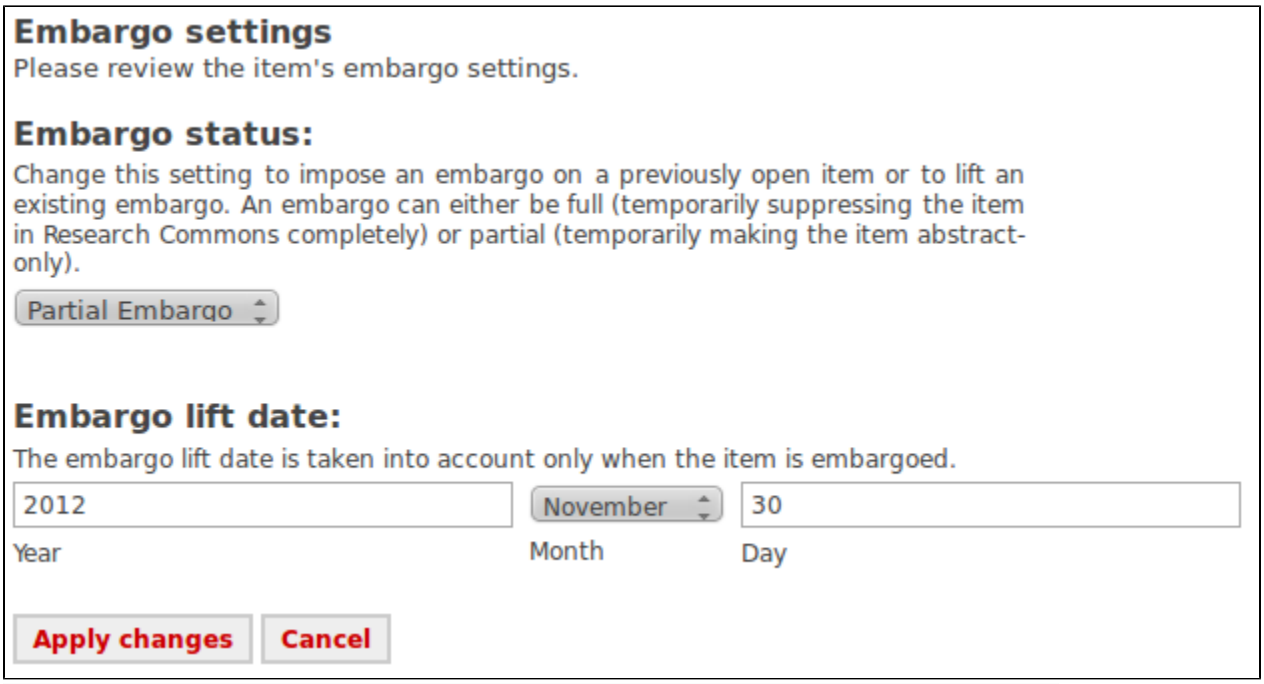


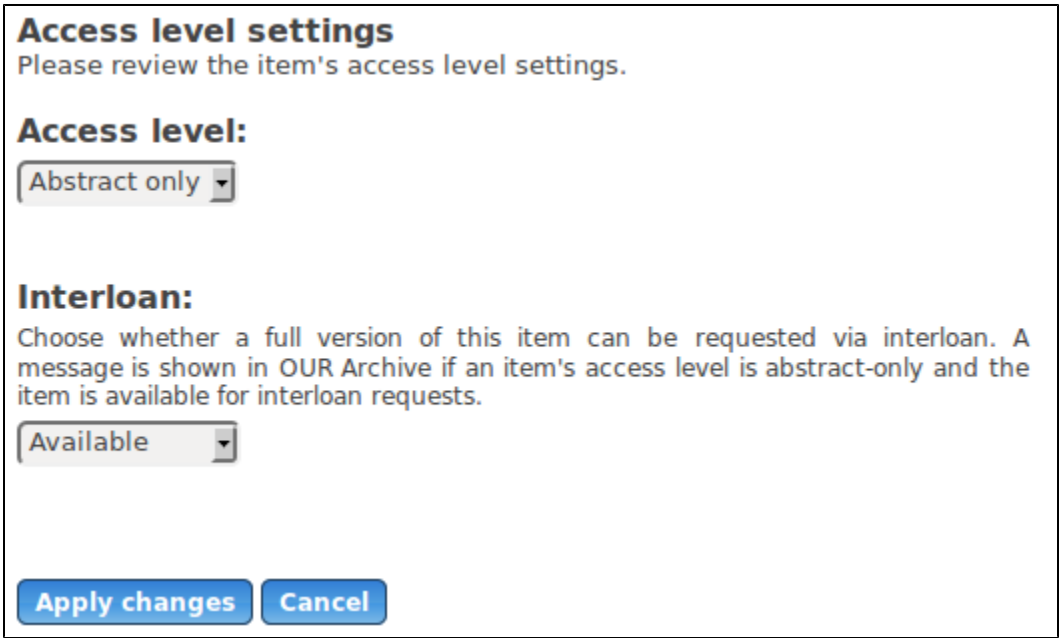


Temporary and permanent restrictions are currently configured on separate screens, though this is mainly for historic reasons. These screens could, and most likely should, be consolidated into one screen; however, it may be desirable to let certain repository staff groups change only temporary or only permanent restrictions.

The first screenshot below shows the embargo screen where files can be fully embargoed only. The second screenshot shows a repository in which full /partial/no embargo can apply. The third screenshot shows permanent restriction settings.







Repository staff have access to a list of all embargoed items via an entry in the "Administrative" menu in the sidebar.

Technical description

The LCoNZ IRR restriction model uses item metadata and resource policies. Separate metadata fields hold the permanent state ("access level"), the embargo type (full vs partial) and the embargo expiry date. Customisations to core DSpace classes ensure that the appropriate resource policies are set up for each item and bitstream whenever the permanent or temporary state changes. Likewise, customisations to core DSpace classes ensure that these resource policies are actually honoured, building on the dark item modifications described by the DSpace@Cambridge folks.