

# Intrallect intraLibrary 3.0: Configuration Guide

Revision: 6  
Created: 4th August 2005  
Last Revised: 13th February 2009  
Contact: [support@intrallect.com](mailto:support@intrallect.com)  
Company: [Intrallect Ltd](#)  
Product: [intraLibrary](#), [Digital Repository](#)  
Copyright: © Intrallect Ltd 2003-2009. All rights reserved.

**This document is made available to support Intrallect's customers and users of Intrallect's software. The text of these documents and the design of the intraLibrary software are both the intellectual property of Intrallect Ltd. Intrallect does not provide this document for any other purpose, and offers no warranty nor accept any liability for its use in any other context.**

## Table Of Contents

|                                       |    |
|---------------------------------------|----|
| 1. Introduction.....                  | 4  |
| 2. Classification Systems.....        | 4  |
| 2.1. Points to Consider.....          | 5  |
| 2.2. Information to Gather.....       | 6  |
| 2.3. Decisions to Make.....           | 7  |
| 2.4. Implementation.....              | 7  |
| 2.5. Future Implications.....         | 8  |
| 3. Metadata Application Profiles..... | 8  |
| 3.1. Points to Consider.....          | 9  |
| 3.2. Information to Gather.....       | 10 |
| 3.3. Decisions to Make.....           | 11 |
| 3.4. Implementation.....              | 11 |
| 3.5. Future Implications.....         | 12 |
| 4. Submission Workflows.....          | 12 |
| 4.1. Points to Consider.....          | 13 |
| 4.2. Information to Gather.....       | 14 |
| 4.3. Decisions to Make.....           | 14 |
| 4.4. Implementation.....              | 18 |
| 4.5. Future Implications.....         | 19 |
| 5. Collections.....                   | 19 |
| 5.1. Points to Consider.....          | 20 |
| 5.2. Information to Gather.....       | 21 |
| 5.3. Decisions to Make.....           | 22 |
| 5.4. Implementation.....              | 23 |
| 5.5. Future Implications.....         | 23 |
| 6. Community Licensing.....           | 24 |
| 6.1. Points to Consider.....          | 24 |
| 6.2. Information to Gather.....       | 24 |

|                                       |    |
|---------------------------------------|----|
| 6.3. Decisions to Make.....           | 25 |
| 6.4. Implementation .....             | 27 |
| 6.5. Future Implications.....         | 27 |
| 7. Appendix A: Workflow Example ..... | 27 |

## 1. Introduction

The purpose of this guide is to introduce the configuration options that should be considered before intraLibrary is installed. The level of understanding required is that of an intraLibrary administrator.

IntraLibrary is very flexible and has various configuration options. In some cases intraLibrary has default settings which many people will find suitable, at least initially. In other cases a decision is required on configuration at the time of setting up intraLibrary, although it may be changed later.

The configuration issues covered in this document are:

- Classification systems (or taxonomies).
- Metadata application profiles.
- Submission workflows.
- Collections.
- Community licensing.

Each section of this guide has sub-sections discussing:

- points to consider;
- information to gather;
- decisions to make;
- how to implement these decisions;
- future implications.

## 2. Classification Systems

Resources within intraLibrary may be discovered by users through searching or browsing. Browsing the library is enabled through the use of classification schemes, or taxonomies, which appear in the left-hand screen in the *Browse* view. Resources are associated with nodes on these classification systems via the *Classification* fields in the resources' metadata. IntraLibrary can support multiple classification systems for classifying resources, and any resource may be classified at any number of nodes in a system or systems.

- For more information on how to browse intraLibrary, see the intraLibrary manual ***Getting Started- Finding and Using Resources***.
- For more information about classifying resources in intraLibrary, see the intraLibrary manual ***Getting Started- Uploading and Cataloguing Resources***.

- For more information about implementing classification systems in intraLibrary, see the intraLibrary *Librarian's Guide*.

Classification systems are presented in intraLibrary as a browsable interface so that a user can navigate through a hierarchy of terms, narrowing the focus gradually until a specific topic is identified. At that stage all resources in that topic area will be revealed in the right-hand screen.

IntraLibrary supports the IEEE Learning Object Metadata ([LOM](#)) standard, and the IMS Learning Resource Meta-data specification that it is based on. In the LOM, classification schemes or taxonomies can be used to classify resources for various purposes, e.g. by discipline (subject), accessibility, educational level, competency or prerequisite. This means that an installation of intraLibrary can support browsing of its resources in a number of different ways, e.g. through a common subject-specific classification (e.g. MeSH for medical resources); a general or universal subject scheme, (e.g. the Dewey Decimal Classification); a local taxonomy based on local curricula, competencies or prerequisites, etc.

### 2.1. Points to Consider

When choosing which classification systems to include, it is important to consider what the purpose of browsing is, in contrast to searching. Searching by keyword or other characteristic tends to happen when the searcher is looking for something specific, whether it is a particular title or topic, or resources that meet specific criteria such as technical format or accessibility options. How to effectively support searching should be considered when developing a metadata application profile, metadata workflow, and quality assurance processes for metadata; see the later sections of this guide for more information.

Browsing tends to be used when a more broad sweep is required, e.g. when someone wants to get an overview of what is available within a particular repository, subject area or educational level. Browsing is often carried out in the early, developmental stages of developing a lecture or course; searches are more targeted to filling specific needs.

For browsing to be effective, the terms in a classification system and their location in the hierarchy need to be intuitive, or at least familiar from repeated use. Developing a classification system from scratch can be very resource intensive and requires some expertise.

It is worth considering the following points:

- Establish who your community of users are. You will need to consider their specific browsing needs, e.g. the subject areas or educational levels they use in their day-to-day work. At least one subject classification is usually necessary.
- Establish whether you will be sharing resources or metadata outwith your own repository. You will need to consider whether to use a widely implemented classification system to ensure interoperability with other services, e.g. if your metadata is being harvested by a national portal, what classification systems do they use? You may require a general or universal subject scheme, such as JACS, Learning Directory or the Dewey Decimal Classification is sometimes needed, alongside a national educational level scheme such as UK Educational Levels ([UKEL](#)).
- If classification systems already exist then it is better to use these than invent new classifications, because of both the resource needed to create and maintain a classification system, and for the sake of ensuring interoperability with other resource provision services.
- As well as being developed for the needs of specific communities, classification systems should ideally be well-maintained by a named group or body; stable; and responsive to community needs.
- In addition to subject and educational level classifications, it may be useful to include classifications for other purposes.

## 2.2. Information to Gather

Gather as much information about existing classification systems as possible. If you do not have a librarian or other information professional on your team, consider asking advice from one in your institution or subject area. In the UK, organisations such as the [JISC-CETIS Metadata and Digital Repositories SIG](#) and Becta's [Vocabulary Bank](#) may also have useful information. Other sources include:

### Universal

Universal classification systems aim to cover the entire possible spectrum of subjects or disciplines. The most commonly used universal systems in libraries are the [Library of Congress Classification Scheme](#), the [Universal Decimal Classification](#) and the [Dewey Decimal Classification](#). Within your community of users, institutional libraries will have preferences which are worth considering. Increasingly, repositories of teaching and learning resources are using universal schemes based on national curricula. In the UK, this includes [JACS](#) for universities and the [Learning Directory Classification](#) (previously known as LearnDirect) for further education.

### Subject

Often, professional organisations establish and maintain classifications. Sometimes these are subject-based but increasingly they may also be competency-based. If your use of intraLibrary has an international perspective then remember that professional organisations are usually national and may differ significantly from one country to another. Examples of subject classifications include [Medical Subject Headings](#) (or MeSH), used internationally for classifying medical resources, and the [Scottish Framework for Social Work Education in Scotland](#) which is both national and competency / learning objective based.

### Educational

It is possible that your use of intraLibrary will span several educational levels and a classification to discriminate between these levels will be useful. Sometimes educational level is considered to be synonymous with qualification. In some simple cases this may be useful but in general it is best to try to aim for a set of educational levels onto which different qualification frameworks can be mapped. Several national qualification authorities have begun to establish such frameworks, such as UK Educational Levels ([UKEL](#)).

## 2.3. Decisions to Make

Decisions to make on classification systems include:

- For what purpose or purposes do you wish to support browsing in your repository?
- Do you need any classification systems to support current or future interoperability with other services?
- Do classification systems already exist that meet your needs, or will you need to develop your own?

These decisions are not irrevocable and it is often advisable to keep things simple at the start; you can learn from your user community and revisit your classification system decisions later. However, once a large number of resources are classified, there will be some work involved in changing your classification systems, so it is best to find a balance between what you need now and what you anticipate needing in the future. For specific details on adding, editing and deleting classification systems, see the intraLibrary ***Librarian's Guide***.

## 2.4. Implementation

There are two ways to implement classification systems in intraLibrary: you can add and edit them manually using the classification system editing tools, or you can import them as [ZThes](#) files. The intraLibrary **Librarian's Guide** describes the process of implementing and maintaining classification systems.

### 2.5. Future Implications

Once a classification system is in place and people have started to use it they will have an expectation that it will persist. They may have set up RSS feeds from classification nodes that interest them, or they may just assume they will always be able to find certain things at certain nodes. In addition, removing or changing a classification system that has been used to classify resources requires decisions about how the classified resources should be handled. Once a large number of resources are classified, there will be some work involved in changing your classification systems, so it is best to find a balance between what you need now and what you anticipate needing in the future. For specific details on adding, editing and deleting classification systems, see the intraLibrary **Librarian's Guide**.

## 3. Metadata Application Profiles

IntraLibrary supports the IEEE Learning resource Metadata ([LOM](#)) standard, and the IMS Learning Resource Meta-data specification that it is based on. This standard is extensive and complex and it is possible to use any subset of it and still conform to the standard. In practice, many organisations and communities choose to use only a part of the whole metadata set; this is known as an *application profile*. Application profiles can define what metadata you use within your repository; they are also necessary for interoperability. Different organisations and communities can see, via application profiles, how other people are using the standard. In addition, an application profile can include fields or properties from other metadata standards where the LOM isn't sufficient for a particular purpose in your repository. For example, intraLibrary supports extending the LOM with a sub-sets of other specialised metadata specifications, e.g. the [NISO z39.87](#) metadata standard for describing technical information about digital images and the [XCRI](#) specification for describing course information.

IntraLibrary comes with some widely used application profiles already built in (see the intraLibrary **Administrator's Guide**). If one of these suits your needs then you can set it to be the default and ignore the remainder of this section.

### 3.1. Points to Consider

The key aspects you need to consider for your application profile are:

- cardinality;
- visibility;
- editability;
- optionality;
- vocabularies.

**Cardinality** allows you to set the number of times a metadata field may be used. In some cases this is limited by the LOM standard. However, in some cases you can decide, for instance, the maximum number of keywords that can be added to a resource's metadata.

**Visibility** allows you to set whether or not a metadata field is visible to users in the *Metadata Editor*. Sometimes fields will be filled automatically using a *metadata template* so there is no reason for people entering metadata to see the field.

**Editability** allows you to set whether a metadata field may be edited by a user in the *Metadata Editor*. This may be important if you have set the system to assign unique identifiers in your *metadata template*; you may want cataloguers to be able to see the identifier, but not to be able to change it.

**Optionality** allows you to select one of five categories for each metadata field: mandatory; recommended; optional; not recommended; not used. For instance, setting certain fields as *mandatory* defines the minimum metadata that must be completed before a resource is published. Setting *Optionality* also allows the *Metadata Editor* to be configured to show different levels of metadata. For instance, some users who are contributing resources may only see the *mandatory* metadata fields when they upload a resource; cataloguers completing the metadata may see *mandatory*, *recommended* and *optional* metadata; while administrators may be able to see all levels of metadata, including fields which are *not used* in the library. This may be useful if they are checking metadata that has been imported with a resource, rather than created within the library. Users can select for themselves which optionality they see in the *Metadata Editor* using a drop-down menu. For this reason, you can over-ride the terms *mandatory*, *optional*, etc. to whatever makes sense for your users. For instance you may like to call *not used* metadata something like *all available fields*; you may like to call *optional* fields *fill in as defined in cataloguing guidelines*.

**Vocabularies** in the application profile define terms for those LOM fields that are populated by standard vocabularies, rather than by free text, identifiers or dates/times. In the *Metadata Editor* and in the *Advanced Search* screen, these vocabularies are available to users in drop-down menus. Each vocabulary may be defined separately. You can select which vocabulary you wish to use in which metadata field in the *Application Profile Editor*. You may also add user-friendly terms or labels to standard vocabulary terms. For instance, in the LOM, some vocabularies are numerical values that are not easily understood by users. You can assign local terms explaining these; the standard LOM terms will still be in the metadata record, but your users will be able to add, see and search by terms they can understand. It is also possible to set whether or not users see the source of a vocabulary you are using in your repository; for instance whether it is a LOM-defined vocabulary, or one defined by a local community for use in a particular field. See the intraLibrary **Administrator's Guide** for more information on defining and editing vocabularies and adding labels.

### 3.2. Information to Gather

The purposes of metadata include:

- supporting accurate, flexible searching;
- enabling users to find out more about a resource before viewing, downloading or using it;
- uniquely identifying resources;
- making statements about the authorship, rights and licensing of a resource;
- allowing other systems to understand how to deliver or process the resource.

With this in mind, defining an application profile also has both an internal and an external function. The internal function is to find a suitable balance between creating high quality metadata to fulfil these purposes adequately, and minimising the effort and resource expended in creating that metadata. The external function is to ensure that, if metadata and resources are to be shared with a wider community, the metadata and the vocabularies used are consistent with, or at least understandable by, those used in that wider community.

For your internal needs, the information to gather is the minimum metadata that would meet the needs of the users of intraLibrary. In addition it is necessary to consider the process by which this metadata will be produced: who will create the metadata; what skills do they have; how much time do they have; how will quality assurance of the metadata be carried out? There is no point setting a field to be mandatory if no-one has the skill or time to

create good quality metadata for that field. Equally, there is no point defining such minimal metadata that the resource is impossible to find in the library, or that leaves your organisation open to problems with mis-attributing resources or violating licensing conditions.

For your external needs, it is necessary to identify the wider communities you may wish to share resources with, and find out if what application profiles are already in use. Some application profiles are defined by the subject domain while others are defined by the geographic region. For example, national application profiles have been agreed in the Canada and the UK by entire educational communities. These application profiles ([CanCore](#) and the [UK LOM Core](#)) are already included out of the box in intraLibrary.

### 3.3. Decisions to Make

The first decision to make is: can you use one of the existing application profiles in intraLibrary? If so then you have no further decisions to make.

If you decide that you need to change a small number of parameters then you can do this easily by editing, or copying then editing, one of the existing application profiles. This makes it simple to change the optionality, cardinality, editability, visibility or vocabularies for each metadata field.

If you decide that you would like a new, named application profile designed for your specific needs then you can use the copy function in the *Application Profile Editor* to copy an existing application profile and edit any or all of its attributes. You can also create an application profile from scratch.

### 3.4. Implementation

To use an existing application profile simply set the desired application profile to be the default in the *Metadata* section of the *Admin* tools. You can also make any number of application profiles *active* in your repository. This means that they will be available for selection for specific groups or collections; they will then be presented with their specific application profile rather than the repository default. This can be useful if, for example, one group or user is working with a type of resource that has specific metadata needs, e.g. an image collection.

To modify the optionality, cardinality, editability, visibility or assigned vocabularies of an existing application profile to make it fit your needs you can edit one of the existing application profiles using the *edit* or *copy* functions in the *Metadata* section of the *Admin* tools.

You may also define a new application profile by clicking the *Add application profile* button in the *Metadata* section of the *Admin* tools.

### 3.5. Future Implications

There are few long-term implications of your initial choice of application profile. It is important to realise that the application profile only affects the *Metadata Editor*, not the metadata itself. Once a resource has been published in intraLibrary there is no record of which application profile was used to create the metadata, unless you choose to name your application profile in the LOM format field. This means that if you choose to change your application profile at any time you can. All subsequent metadata will conform to the new application profile (ensuring that, for example, a set of fields now defined as mandatory will always be present).

The other implication of application profiles relates to subsequent exposure or sharing of your metadata. If you are exposing metadata for harvesting or external searching, you may like to consider ensuring that you expose your application profile so that external services and systems understand what to expect from your metadata records. In this case, good version control on your application profiles and ensuring that you do declare what application profile you have used within your metadata records may be helpful.

## 4. Submission Workflows

A submission workflow is the process through which a resource is uploaded into intraLibrary, metadata created, classification, checking, reviewing and other quality assurance steps are carried out, and the resource is published (made available to all intraLibrary users). The default workflow allows a single individual to carry out all these actions in a single stage. If that workflow is suitable then no configuration decisions need to be made and this section is not relevant.

On the other hand, intraLibrary supports much more complex workflows and these

workflows are very configurable. In addition, different workflows can be used by different groups of users, and individuals can belong to multiple groups, and they can play different roles in each different group. The remainder of this section describes configuration for these more complex submission workflows. The details of allocating users to groups, roles to users, and workflows to groups is dealt with in the intraLibrary **Administrator's Guide**. This section discusses how to design and configure a workflow - a prerequisite before the stages described in the intraLibrary **Administrator's Guide**.

### 4.1. Points to Consider

A workflow is normally split into a sequence of stages (though a single stage workflow is also possible). Each stage may include one or more processes. Each process has one or more user roles associated with it and a set of permitted actions. At the transition between one stage and another a resource may have its published state changed, i.e., it may become published (so that it is viewable by all users of intraLibrary) or unpublished (so that it is hidden from all users except the group engaged in the workflow). At any of these transitions, email alerts can be sent to each of the users with a role in the next stage.

A useful place to start when considering submission workflows is to identify who will be involved in the entire submission process. There is little to be gained by creating five different stages if only two people will carry out the work - although one option is to use this approach to differentiate the different roles that a single person might have to take on.

Another point to consider is the process of decision-making through the submission workflow. A decision is a point at which a choice must be made. Some typical questions demanding choices might be:

- How much metadata might it be reasonable to expect the resource contributor to create?
- What metadata fields would you like people in different user roles to be able to see and/or edit?
- What is the minimum amount of metadata required before a resource could be published (made available to all users)?
- Who has the authority to create rights metadata?
- Who has the authority to publish a resource?
- What action needs to be taken if a resource fails at a review stage?
- How much metadata should be expected when the full submission process is completed?

- Might it be necessary to overwrite a resource with a new version during submission?

## 4.2. Information to Gather

As submission workflows are exclusively an internal consideration there is no requirement to consult more widely than the team involved in the workflow. There is, however, a growing body of work on how metadata quality is affected by workflow and it is worth being aware of the lessons from this work. See for example:

- Currier, S., Barton, J., O'Bierne, R., Ryan, B. (2004). Quality assurance for digital learning resource repositories: issues for the metadata creation process. ALT-J Vol. 12, no. 1 (Mar. 2004), pp. 5-20. (Or see [this article](#) for a summary).
- Metadata Workflow Investigation <http://mwi.cdlr.strath.ac.uk/Index.htm>.

## 4.3. Decisions to Make

To design a workflow it is necessary to decide on stages, processes, actions and roles and combine them, for example in a simple sketch. Starting from the widest viewpoint decide how many stages you want and name them. Narrowing the view slightly, decide if each stage should contain one or more processes. Limiting a stage to a single process keeps it simple and easy to follow the progress of a resource through the workflow. On the other hand, when there are many processes to be carried out, each taking some time to be completed, the entire workflow will be completed sooner if some of the processes can be carried out in parallel during a single stage.

For each process, decide on the roles that should be used to undertake the task. You are free to give titles to these roles.

Then decide which actions someone in that role needs to be able to carry out. The set of possible actions is limited to those listed in the table below.

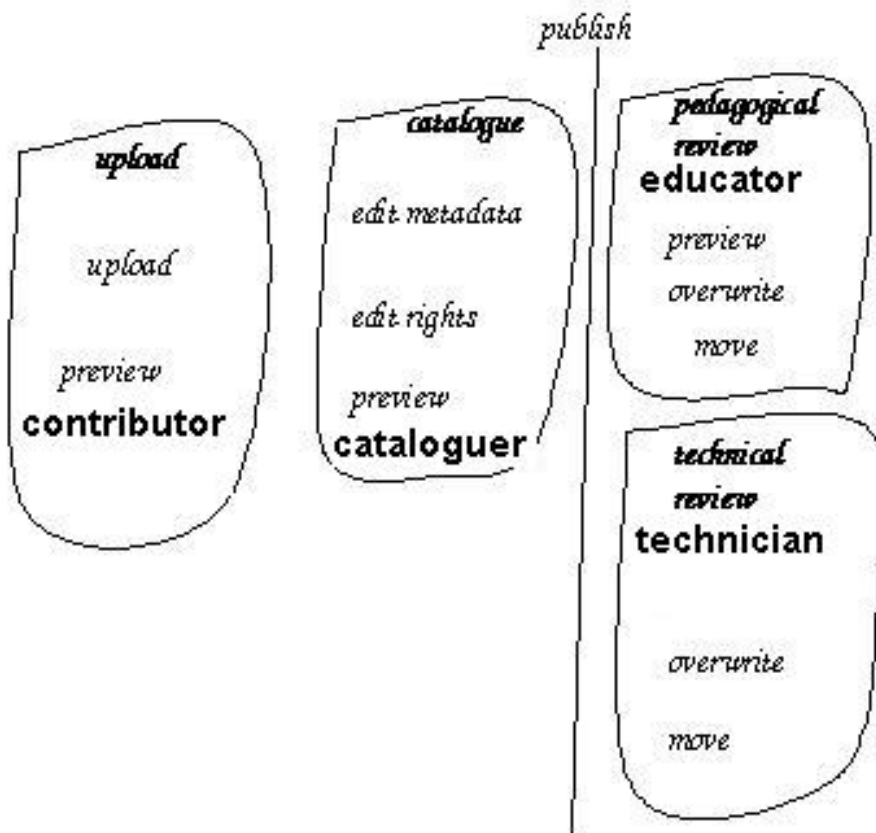
| Action             | Workflow XSD Element | Description  | Additional Parameters |
|--------------------|----------------------|--|-----------------------|
| Change collections | change_collections   | Modify which collections the resource is a member of. User can put the resource in any |                       |

|                        |                         |   |  |
|------------------------|-------------------------|---|--|
|                        |                         | collections they have contribute access to.   |  |
| Delete                 | delete                  | Remove a resource completely from the system.   |  |
| Edit metadata          | edit_metadata           | Metadata / classification editor configured by the chosen Application Profile, and an optional metadata subset.   | Metadata Subset. See <a href="#">Metadata Subsets</a> section. |
| Edit rights            | edit_rights             | Licence (rights management) editor.   |  |
| Export                 | export                  | Package export functions.   |  |
| Import from filesystem | import_from_file_system | Import a file or content package from the group's folder on the server filesystem.  |  |
| Link resources         | link_related_objects    | Link a resource to any other resource by populating the LOM Relation fields for both resources. All resources must have the correct LOM field and vocabulary assigned in the relevant application profile, and must have title and LOM identifier fields populated. |  |

|                             |                              |   |                                      |
|-----------------------------|------------------------------|---|--------------------------------------|
| Move resource               | move_object                  | Send the resource to another stage of the workflow, with comments attached. Typically used at review stages.  | Stage ID                             |
| Overwrite                   | overwrite_resource           | Upload a new version of a resource to the system. New version has same metadata as old resource. The exception is if a new title has been entered at upload or imported in a content package. |                                      |
| Preview                     | preview                      | Package/file preview.   |                                      |
| Return                      | return                       | Puts the resource back into the workflow and published state it had before it entered the current workflow.   |                                      |
| Save manifest to filesystem | save_manifest_to_file_system | Write a package manifest to a folder on the server filesystem. Folder location is a property of the workflow process.   | Path of folder on server filesystem. |
| Save to filesystem          | save_to_file_system          | Write a content package to a folder on the server filesystem.   | Path of folder on server filesystem. |

|               |               |  |  |
|---------------|---------------|--|--|
| Transfer      | transfer      | Move the resource into a different workflow stage or different group. User can move the resource to any group/stage they have access to. |  |
| Upload        | upload        | Upload or add a new resource to the system.  |  |
| View Metadata | view_metadata | Metadata preview.  |  |

Once all decisions have been made you can implement the design.



### 4.4. Implementation

To implement a workflow design the conceptual design needs to be turned into an XML file which is valid in terms of the intraLibrary workflow schema:

[http://www.intrallect.com/xsd/intralibrary\\_workflow\\_v0p5.xsd](http://www.intrallect.com/xsd/intralibrary_workflow_v0p5.xsd)

An example of a workflow in XML form is given in [Appendix A](#).

The workflow has a number of elements:

#### **workflow**

A workflow is required to have a *name* and a *description*. It also must have at least one stage, although there is not limit to the number of stages possible.

#### **stage**

A stage must have a *name* and a *description*. It also must have at least one process, though the number of processes is not limited. Each stage may have up to one each of four properties: *changePublishState* which means that when the stage is completed the resources published state will change, either from published to unpublished or vice versa; *completeOnAny* which means that completion of any one of the processes in a stage will define the stage as having been completed - the default is that all processes must be completed before the stage is considered completed; *alertObjectOwner* which means that the owner of a resource receives an email when the workflow stage is completed for that resource; *emailAddressesToAlert* which defines email addresses that will be alerted when a resource has passed through a stage.

#### **process**

A process must have a *name* and a *description*. It also must have at least one *action*, though the number is not limited. Each process may also have a number of optional elements:

##### **role**

A process may have none, one or many roles assigned to it.

##### **ownerAccess**

The resource owner, by default, has no access to each process. By declaring *ownerAccess* for a process the owner can carry out the actions in the process.

##### **metadataSubset**

A metadata subset can be specified to identify the metadata associated with a process. There can be none, one or many metadata subsets but each subset must have a *name*. Metadata subsets may be defined and edited in the *Metadata* section of the *Admin* tools. The subset must be defined in intraLibrary prior to the workflow being

imported.

### **stageToMoveTo**

When one of the actions is *move resource* one occurrence of stageToMoveTo is required and should define the stage to which a resource will be moved, specified by a stageposition with the first stage numbered as 1.

### **action**

Every action must have a type and the type should be one of:

- change collections
- delete
- edit metadata
- edit rights
- export
- import from file system
- link resources
- move resource
- overwrite resource
- preview
- return
- save manifest to file system
- save to file system
- transfer
- upload
- view metadata.

## **4.5. Future Implications**

In general, workflows have no impact on resources and metadata in the repository. Once you have a workflow in place, and there are resources in your repository being worked on within the workflow, you will need to move any resources already in the workflow being changed into the new workflow. From intraLibrary 3.0, a Librarian user can move resources between workflows in bulk using their Basket, so, while there are resource implications, this will not be too resource intensive.

## **5. Collections**

Collections are used to define sets of resources with common attributes such as those provided by particular publishers or institutions, resources of a particular media type, or resources with particular usage restrictions. The repository can have many collections which can have different levels of access for different people. It is also possible to set various properties for collections; this will be discussed in the following section. If you intend to have all the repository's resources visible to all users then this section may not be relevant. Consult the intraLibrary **Administrator's Guide** for more detail on collections and groups.

## 5.1. Points to Consider

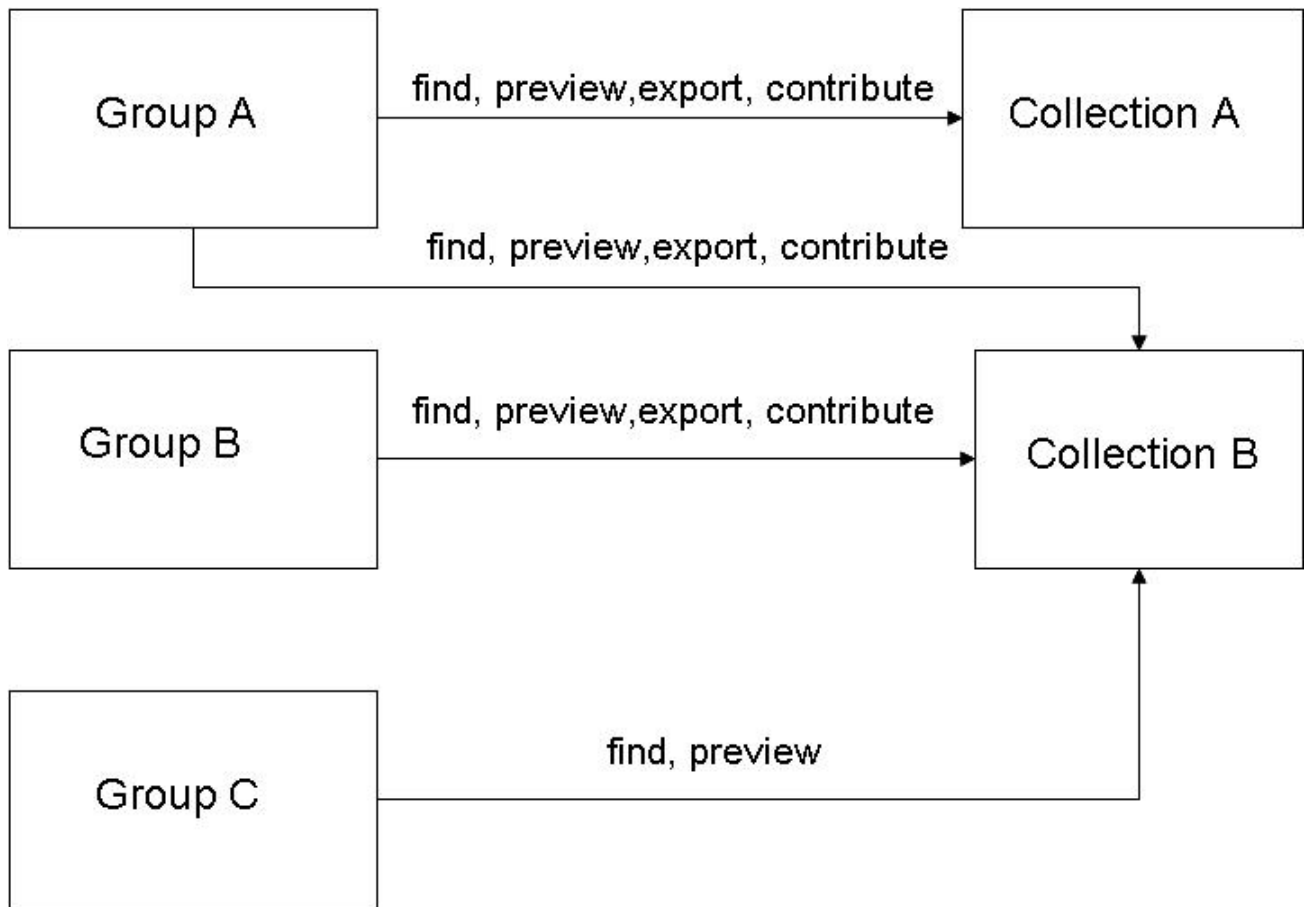
It is important to note the strength of the relationship between groups and collections. For the user, access to collections is determined by which groups the user is in. In addition, contribution features can be controlled for resources entering a collection. In particular the application profile, licence model and classification systems used can be set based on what collection a resource is in. The classification systems seen by a user will also depend on which collections the user has access to.

The relationships between collections and groups are defined through the following permissions:

| Permission | Description   |
|------------|---|
| Find       | Allow users in the group to find resources in the collection using search and browse  |
| Preview    | Allow users in the group to preview resources in the collection. NB: for individual files (rather than content packages) this permission effectively allows users to download the resource                                    |
| Export     | Allow users in the group to download the resource, metadata and public URL of the resource  |
| Contribute | Allow Contributor users in the group to add resources and edit metadata in accordance with their role in the group  |
| Annotate   | Allow users in the group to add comments, star ratings and tags to resources in the collection. NB: even if this permission is switched off, the users will still be able to view annotations for resources in the collection |

The figure below clarifies what relationships can be made between groups and collections. Note that the user type and group role takes precedence over the collection permission.

For example: A contributor in group B would have permission to find, preview, export and contribute to collection B and a user in group B would be able to find, preview and use published resources in collection B. However a contributor in group C would only be able to find and preview resources in collection B.



## 5.2. Information to Gather

Although you can put each resource into one or more collections it is useful to think about collections as a way of dividing up your repository into different sections. Reasons why you might want to divide up the repository include:

- You may want to have some resources that can not be seen by all users of the repository.
- You may wish to have different types of metadata and classification systems for certain

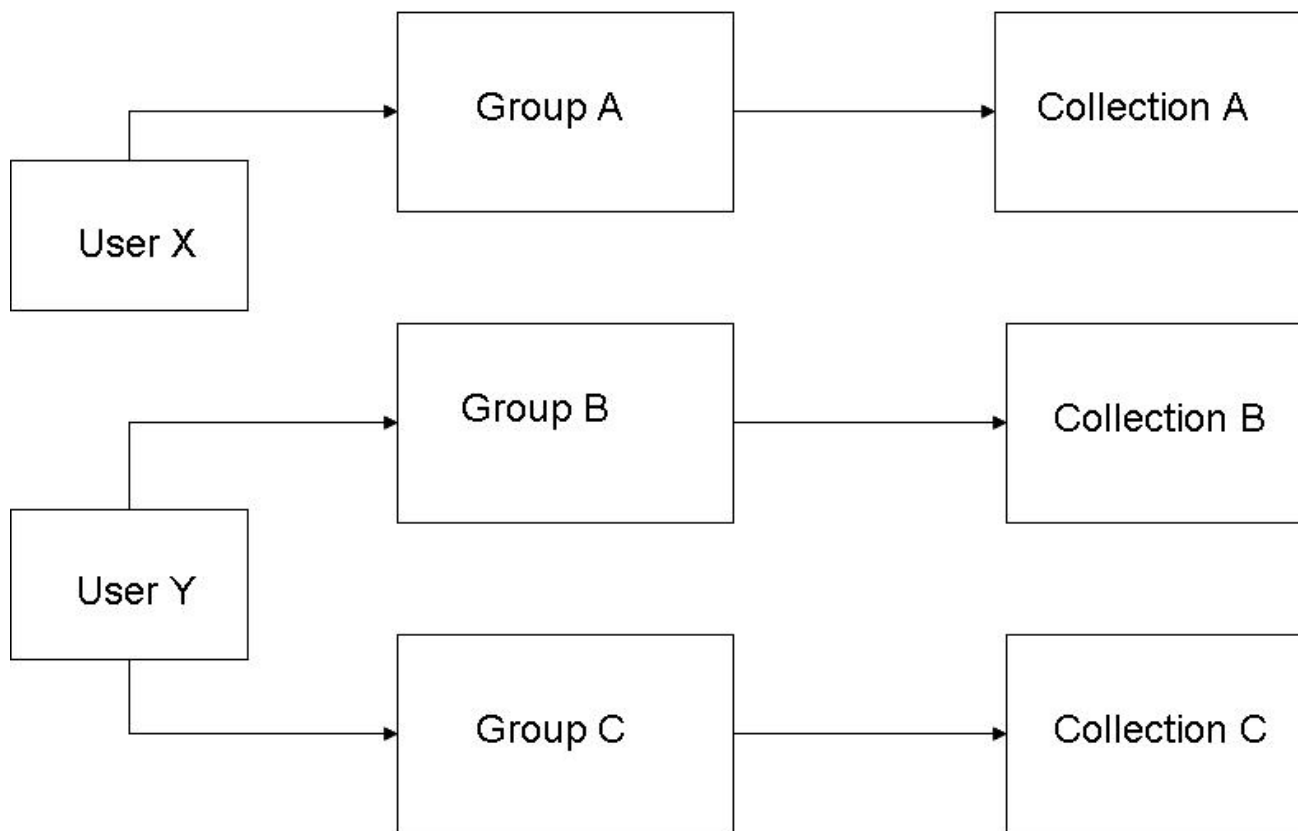
types of resources.

- You may wish to make some (but not all) resources in a repository available to be used by external systems for metadata harvesting or through federated searching.
- You may wish to make some (but not all) resources in a repository externally accessible via an RSS feed.

These can all be implemented using intralibrary. As an initial step, it is useful to consider all the users of the repository, what resources you would like each of them to see and what actions you would like them to be able to perform. Then you need to consider how this can be achieved by putting these users into groups and relating them to collections.

### 5.3. Decisions to Make

The administrator needs to decide upon the best way to link groups and collections. The example shown above is one way in which groups could be related to collections. In practice, for a given situation it may be possible to implement a solution in different ways using a different combination of groups and collections. The simplest way to relate groups to collections is on a one-to-one basis as shown in the diagram below. Using this strategy, you could choose which collections a user has access to by making them members of the appropriate groups.



## 5.4. Implementation

Implementing collections in intraLibrary is achieved using the collections section in the admin tools. Using this page it is possible to create collections, edit properties of a collection and set access permissions for a collection.

## 5.5. Future Implications

The use of collections allows the administrator to have a lot of control over who has access to a collection and the features of resources which are added to the collection. It is possible to change which collections a resource is in, and from intraLibrary 3.0, Librarian users can move resources between collections in bulk using their Basket. The main consideration is with the choice of classification systems, application profiles and licences for a collection and the long term issues with these are discussed in other parts of the manual.

## 6. Community Licensing

IntraLibrary's community licensing capabilities cover two main areas: the conditions under which users agree to use the repository (described as agreements below) and the conditions under which they agree to use each resource (described as licences).

Adopting any of the community licensing is optional. By default intraLibrary has no agreements and no licences in place. If this suits your configuration then there is no need to read this section.

### 6.1. Points to Consider

First, consider the need for an agreement for users. This could require users to agree to certain terms and conditions when they use the repository. It may be necessary to have different terms and conditions for different user roles. For example, a contributor of resources might be asked to ensure that they have the right to submit resources while a user who is not a contributor might only be asked to respect the terms and conditions associated with each resource. In some circumstances it might be appropriate to use paper-based agreement forms. However, if online agreements are suitable then intraLibrary can provide a number of options.

Secondly, and completely separately from an agreement about the terms and conditions associated with using the repository, it is necessary to consider the licensing conditions under which copyright owners will be able and willing to make resources available for others to use. The most important aspects of these licences are:

- The presence of a licence gives contributors the reassurance that they know exactly the conditions under which their contributions may be used.
- It is perfectly practical to have several licences with different conditions so that contributors can select the licence conditions that are most suitable.

### 6.2. Information to Gather

For agreements on the use of the repository there is little need to gather information. It is helpful to gather the views of the repository community and it may be useful to examine similar agreements from other repositories.

In order to select suitable licences it is worth gathering information on existing licences which may be appropriate. The benefits of using pre-existing licences is that the repository users may already be familiar with the terms of an existing licence. Examples of licences which may be appropriate include Creative Commons licences (<http://www.creativecommons.org>) and other forms of public licence.

You may find it useful to refer to some of the work Intrallect has done in this area:

- [JISC Digital Rights Management Study](#).
- [IPR in International E-Learning Programmes](#).
- [Creative Commons in the Common Information Environment](#).

### 6.3. Decisions to Make

To set up agreements in intraLibrary it is necessary to decide on the exact text of an agreement and also whether it will apply to all users or only to users in one of the specific intraLibrary roles (guest, contributor, librarian, administrator). You will need to decide whether the user should be required to indicate acceptance of the agreement only on the first login, or every time they login. In addition, each user may receive an email confirmation with the full text of the agreement.

In order to set up a single licence, or a set of licences from which contributors can select, it is necessary to make a number of decisions. First, consider the uses that should be permitted for the resource. The available uses can be usefully grouped into four sections:

- Rendering
  - display - the act of rendering the resource on a visual device.
  - print - the act of rendering the resource onto paper or hard copy form.
  - play - the act of rendering the resource into audio/video form.
  - execute - the act of executing the resource. (For example, machine executable code, Flash or Java)
- Transport
  - give - the act of allowing the resource to be given away in perpetuity without exchange of value.
  - sell - The act of allowing the resource to be sold in exchange of value.
  - lease - the act of allowing the resource to be made available for a fixed period of time then returned, or access ceased (for exchange of value). During this period, the resource is only available to the lessee. Temporal constraints must be defined.
  - lend - the act of allowing the resource to be made available for temporary use then

- returned, or access ceased (without exchange of value). During this period, the resource is only available to the lendeer. Temporal constraints must be defined.
- **Modification**
    - excerpt - the act of extracting (replicating) unchanged parts (or all) of the resource for reuse in another resource.
    - annotate - the act of adding notations/commentaries to the resource creating a new resource.
    - aggregate - the act of using a resource (or parts of it) as part of a composite work or collection.
    - modify/adapt - the act of changing parts of the resource creating a new resource.
  - **Utility**
    - move - the act of allowing a resource to move between data storage devices.
    - duplicate - the act of making an exact copy of a resource between data storage devices.
    - backup - the act of making copies of a resource for the purpose of guarding against the loss of the original due to accident or catastrophic media or equipment failure.
    - save - the act of saving a copy of a resource to permanent storage.
    - restore - the act of allowing the conversion of a backup copy into a usable copy in a controlled manner.
    - verify - the act of allowing authorisation to check the authenticity of an asset.
    - uninstall - the act of allowing for the removal from or disabling of a resource in a data storage device.
    - delete - the act of deleting a copy of the resource

It may be easy to decide on groups of uses, for example, it is common that permission will be granted for all rendering and utility uses. It may also be the case that all modification and transport uses will also be permitted but it is more likely that you will want to permit some and deny others.

Once you have decided on the permitted uses you will also want to consider possible constraints on these uses. The constraints that can be supported by the intraLibrary licensing model are:

#### Attribution

Whenever the resource is used its creators should be recognised and attributed.

#### Non-commercial

All the permitted uses are only permitted for non-commercial purposes.

#### ShareAlike

This constraint applies only if some transport uses have been permitted. It requires anyone transferring the resource to someone else to do so under the same conditions

as you are making it available to them.

Once you have decided on the set of permitted uses and constraints you (or your lawyer) should create a full text licence matching these conditions and make it available on a web site. It is worth comparing your uses and constraints with those provided by the set of Creative Commons licences as one (or more) of these licences may meet your needs. You can then include the URL of your full text licence in your licence model metadata. It is usual to insist that the licence should be kept with the resource whenever it is transported.

### 6.4. Implementation

Implementing agreements in intraLibrary is achieved filling in the form in the agreements section of the admin tools.

Implementing licences in intraLibrary requires a detailed setup which should be carried out by contacting Intrallect to discuss specific requirements as described in the previous section.

### 6.5. Future Implications

When you establish agreements and licences you are establishing a form of contract. Although you can change that contract at some time in the future for subsequent uses of resources in your repository you will be expected to honour any previous agreement or licences.

It should be remembered that if you allow transport uses then you will be unable to keep track of who has access to your resources. The trade-off from this lack of control is that your resources may be widely adopted because you have made them easy to use and share.

## 7. Appendix A: Example of a Workflow

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<workflow xmlns="http://www.intrallect.com/xsd/intralibrary_workflow" xmlns:
xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.intrallect.com/xsd/intralibrary_workflow http://www.intrallect.com/xs
name="Quality Assurance Workflow 2.0" description="Workflow defined for learning resource deve
```

```
<!-- ***** Stage 1 ***** -->
<stage name="Contribute" description="Upload a learning resource and publish">

    <process name="Contribute" description="Upload a learning resource and publish">
        <actions>
            <delete/>
            <edit_metadata metadataSubset="EducationalMetadata"/>
            <edit_rights/>
            <preview/>
            <upload/>
        </actions>
        <role name="Content Contributor"/>
        <role name="Submission Manager"/>
    </process>
    <changePublishState/>
</stage>
```

```
<!-- ***** Stage 2 ***** -->
<stage name="Catalogue" description="Enter complete metadata">
    <process name="Catalogue" description="Enter more details metadata">
        <actions>
            <edit_metadata/>
            <preview/>
            <view_metadata/>
            <move_object stageToMoveTo="1"/>
        </actions>
        <role name="Metadata Cataloguer"/>
        <role name="Submission Manager"/>
    </process>
</stage>
```

```
<!-- ***** Stage 3 ***** -->
<stage name="Review" description="Check and revise, or reject metadata">
    <process name="Review" description="Check and revise, or reject metadata">
        <actions>
```

```
        <edit_metadata/>
        <preview/>
        <view_metadata/>
        <move_object stageToMoveTo="2"/>
    </actions>
    <role name="Reviewer"/>
    <role name="Submission Manager"/>
</process>
</stage>
</workflow>
```