



Intrallect intraLibrary 2.9: Integration Guide

Revision: 4

Created: 1st August 2005

Last Revised: 16th July 2007

Contact: support@intrallect.com

Company: [Intrallect Ltd](#)

Product: [intraLibrary](#), [Digital Object Repository](#)

Copyright: Copyright Intrallect Ltd 2003-2007. 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 offer no warranty nor accept any liability for its use in any other context.

Table Of Contents

1. Introduction.....	3
2. Metadata Harvesting	3
2.1. Opening Collections for Harvesting	4
2.2. Entry Point	4
2.3. Metadata Formats.....	4
2.4. Testing	4
2.5. Modifying the Metadata.....	4
3. Search and Retrieve Service	5
3.1. SRU	5
3.2. SRW.....	5
3.3. Queries	6
3.4. Additional Search Indices	6
3.5. Record extensions	7
3.5.1. Record Metadata Schema.....	8
3.5.2. Intrallect's Review Extensions	9
3.5.3. Intrallect's Package Extensions.....	10
3.6. Metadata formats	12
3.7. Access Control.....	12
3.8. Modifying the Metadata.....	13
3.9. Conformance	13
4. File transfer.....	13
5. IMS DRI	14
6. Repository Java-API.....	14
7. External Authentication.....	15
8. Future Developments	16
9. References	16

1. Introduction

This guide is for those responsible for managing software systems which will need to integrate with intraLibrary.

The following integration points are described in this guide.

Metadata Harvesting

The metadata harvesting interface, based on the Open Archive Initiative - Protocol for Metadata Harvesting ([OAI-PMH](#)), allows the metadata from intraLibrary to be harvested in XML form for use in other systems such as catalogues, aggregators and archives.

Search and Retrieve

The Search and Retrieve URL (SRU) and Search and Retrieve Web-Service (SRW) are related specifications, which both enable external systems to search intraLibrary and retrieve the search results.

File Transfer

How organisations can automate the upload of packages and files.

Digital Repositories Interoperability

This interface offers a subset of the repository services defined in the [IMS Digital Repositories Interoperability Specification](#).

Repository Java-API

Deeper access to the underlying functionality of intraLibrary, used for writing custom integration tools and extensions.

Authentication

Various ways intraLibrary can share and exchange user authentication information with other systems.

2. Metadata Harvesting

An IntraLibrary repository can expose metadata for harvesting by other systems, using the Open Archive Initiative - Protocol for Metadata Harvesting (OAI-PMH). This section describes how intraLibrary supports OAI-PMH. For full details of the protocol visit the [OAI web site](#). IntraLibrary supports version 2.0 of OAI-PMH.

2.1. Opening Collections for Harvesting

Any intraLibrary repository may contain multiple collections of digital objects. The collections in an intraLibrary repository are closed by default, which means records contained in the collection are not harvestable. An administrator-class user of intraLibrary can open the collection for harvesting by changing a single property of the collection. Individual collections are exposed as "sets" under OAI-PMH.

2.2. Entry Point

The entry point for harvesting metadata under OAI-PMH from intraLibrary is the root URL for the intraLibrary installation, with "/IntraLibrary-OAI" appended to it:

`http://<intralibrary-root>/IntraLibrary-OAI`

2.3. Metadata formats

The metadata formats that can be harvested from intraLibrary are summarised in the following table.

Name	Prefix	Specification
Dublin Core	dc	http://dublincore.org/documents/dces
IMS Metadata 1.2	imsmd	http://www.imslobal.org/metadata/index.html
IEEE LOM	lom	http://ltsc.ieee.org/wg12/materials.html
ODRL (version 1.1)	o-ex	http://odrl.net/

2.4. Testing

You can test the harvestability of a repository using the [OAI Registry Explorer](#). This allows you to send commands to intraLibrary. Type in the URL including `/IntraLibrary-OAI` and then you can select commands such as "Identify", "List Metadata Formats", etc. Note that for some commands you need to use parameters.

2.5. Modifying the Metadata Returned

IntraLibrary can be configured to overwrite parts of the metadata harvested under OAI-PMH. In the "General Configuration" section of the "system" page there is an option to select a metadata template to apply to harvested metadata. The values and macros in the selected template will overwrite the values stored in the relevant fields of each record returned in a harvest operation.

Possible uses for the feature include:

- consistently exposing useful properties, such as PublicURLs in specific metadata fields
- blanking the contents of certain fields for privacy or security reasons
- ensuring consistency in important statements, such as those describing copyright or usage policy

3. Search and Retrieve Service

IntraLibrary supports two related specifications for search and retrieve services, both open specifications published by the Library of Congress:

- [Search and Retrieve URL \(SRU\)](#)
- [Search and Retrieve Web-Service \(SRW\)](#), also known as "SRU Web Services"

Both these specifications support searching a remote catalogue and retrieving search results. They use the same query language, Common Query Language (CQL) and similar XML formats for the results of an "explain" and "query" operation.

3.1. SRU

Search and Retrieve URL (SRU) is a REST-like interface. It can be called with a simple HTTP GET request, with parameters specified in the [query string](#) of the URL, and responds with an XML document. The entry point for the SRU service is:

- `http://<intraLibrary-root>/IntraLibrary-SRU`

3.2. SRW

The entry point for the SRW service is:

- `http://<intralibrary-root>/services/SRW`

The SOAP binding has been implemented as follows, based on guidance from the Library of Congress team and the [standard WSDL for SRW](#):

- Service Style: document/literal
- Message Style: inline with no multi-refs
- SOAP version: 1.1

3.3. Queries

CQL

Queries in SRU and SRW are defined in [Common Query Language \(CQL\)](#). It is very straightforward to create simple queries in CQL, but it is also possible to create complex queries, with multiple constraints. IntraLibrary supports "Level 1" of CQL.

Searches can be constrained by specific metadata fields using search indices in the [Dublin Core context set](#). There is currently no standard context set for IEEE LOM metadata. Support for some [additional search indices](#) has been added, which are described in the next section.

Some simple queries:

- fish
- dc.title=metadata
- dc.creator="john smith"
- dc.keyword=snow
- dc.subject="Natural history of organisms"

3.4. Additional Search Indices

IntraLibrary supports selected indices from additional CQL "context sets". It is anticipated that support for more context sets will be added in future versions of IntraLibrary.

Record Metadata Context Set

The [Record Metadata Context Set](#) includes a range of "indices" covering various

information that might recorded in the metadata of a digital object. The indices supported by intraLibrary are listed in the following table.

index name	description
rec.collectionName	Name of a collection which contains the record. This is set by editing the name of a collection in the "admin tools" interface of intraLibrary.
rec.collectionIdentifier	Identifier for a collection which contains the record. This is a new property of a collection, which can be set by editing the properties of a collection in the "admin tools" interface of intraLibrary.

The index *rec.collectionName* and *rec.collectionIdentifier* can be used as part of a query to specify particular collections. For example a query can:

- constrain a results set to objects in one collection only
 - dc.title = "cat" and rec.collectionName = "e-Prints"
- return the entire contents of a collection
 - rec.collectionIdentifier = "uk.ed.epr1"
- obtain results from more than one collection
 - dc.title = "cat" and (rec.collectionName = "images" or rec.collectionName = "learning objects")

NB: The *rec.collectionName* and *rec.collectionIdentifier* indices replace the "collection" context set which was previously made available for use with intraLibrary 2.8, and is now no longer available.

3.5. Record Extensions

IntraLibrary supports a range of extensions to record metadata returned by a SRU/SRW request. These are needed to support various additional functionality requested by various organisations using intraLibrary.

Where possible Intrallect has adopted existing extensions, such as parts of the [Record Metadata Schema](#). Where Intrallect has added its own extensions, they have been assigned a unique namespace using [Intrallect's own "authority string"](#).

3.5.1. Record Metadata Schema

Record Metadata Schema

The [Record Metadata Schema](#) includes a range of extension elements covering various additional information that might be recorded in the metadata of a digital object. The extension elements supported by intraLibrary are listed in the following table.

element	description
modified	date/time when the record was last modified
created	date/time when the record was created

The following example shows how the "rec" extensions may appear in the context of a record returned as part of an SRU/SRW response. The "rec" prefix is mapped to the XML namespace "http://srw.o-r-g.org/schemas/rec/1.0/".

```
<record>
  <recordSchema>info:srw/schema/1/dc-v1.1</recordSchema>
  <recordPacking>xml</recordPacking>
  <recordData>
    <srw_dc:dc>
      <dc:title>This is a Sample Record</dc:title>
    </srw_dc:dc>
  </recordData>
  <recordPosition>1</recordPosition>
  <extraRecordData>
    <rec:record>
      <rec:history>
        <rec:modification>
          <rec:modified>2006-12-09T12:10:00</modified>
          <rec:created>2004-11-09T13:11:21</created>
        </rec:modification>
      </rec:history>
    </rec:record>
  </extraRecordData>
</record>
```

3.5.2. Review Extensions

Intrallect has defined a set of elements to expose statistical information about the reviews/comments that have been made on objects in the repository. The "review" extensions are designed to be included in the "extraRecordData" element of the SRU/SRW response, mapped to the XML namespace "info:srw/extension/13/review-v1.0".

Element Name	Description
meanStarRating	the mean of all the star ratings (0-5) of the digital resource made by users of the repository
numberOfStarRatings	the total number of star ratings made on the digital resource by users of the repository
numberOfComments	the total number of comments made on the digital resource by users of the repository

The following example shows how the "review" extensions may appear in the context of a record returned as part of an SRU/SRW response. The "review" prefix is mapped to the XML namespace "info:srw/extension/13/review-v1.0".

```
<record>
  <recordSchema>info:srw/schema/1/dc-v1.1</recordSchema>
  <recordPacking>xml</recordPacking>
  <recordData>
    <srw_dc:dc>
      <dc:title>This is a Sample Record</dc:title>
    </srw_dc:dc>
  </recordData>
  <recordPosition>1</recordPosition>
  <extraRecordData>
    <review:meanStarRating>3.5</review:meanStarRating>
    <review:numberOfStarRatings>10</review:numberOfStarRatings>
    <review:numberOfComments>17</review:numberOfComments>
  </extraRecordData>
</record>
```

3.5.3. Package Extensions

Intrallect has defined a set of elements to expose information about the packaging of digital objects in the intraLibrary repository system. The "package" extensions are designed to be included in the "extraRecordData" element of the SRU/SRW response, mapped to the XML namespace "nfo:srw/extension/13/package-v1.0".

Element Name	Description
packageType	The type of content package that is stored. A sample vocabulary for this field is given below.
packageTypeVersion	The version of the content packaging specification the package conforms to
packageDownloadLocator	A locator, typically a URI, but could be another resolvable identifier of any kind, from which the package can be downloaded, with the appropriate authorisation
packageManifestLocator	A locator, typically a URI, but could be another resolvable identifier of any kind, at which the manifest file of the package can be accessed in the context of the package as a whole. This allows a structured package to be delivered by a remote system, such as a Learning Management System (LMS), whilst the content itself remains in the repository. There are various scenarios where this could be valuable, including: <ul style="list-style-type: none">• display a structured IMS or SCORM package inside a VLE/LMS, whilst it remains in the repository• making a link to a particular page inside a content package, such as an internal navigation page or index• handing the responsibility for playing/previewing particular kinds of packages to appropriate services or engines, such as<ul style="list-style-type: none">• an IMS QTI player• an IMS Learning Design engine• a SCORM player or IMS Simple Sequencing engine• an IMS Common Cartridge player
packagePreviewLocator	A locator, typically a URI, but could be another resolvable identifier of any kind, which resolves to the built in "public

	<p>preview" function of intraLibrary. The behaviour of the public preview depends on the type of objects that is stored:</p> <ul style="list-style-type: none"> • single file: the file is returned for display by the browser or appropriate browser plugin or helper application • multi-file package: a frameset with the browsable structure of the package in the left-hand frame, and the first "page" of the package in the right-hand frame • web (URL) reference object: forwards to the stored URL • record of a "physical" object: a page displaying the stored location of the physical object
--	--

The following table defines a sample vocabulary for the "packageType" element. The current version of intraLibrary supports a subset of these package types and terms, as indicated in the table.

Term	Usage	Supported
ims-common-cartridge	zip file structured according to the IMS Common Cartridge specification (when released)	No
ims-cp	zip file structured according to the IMS Content Packaging specification, in its generic form	Yes
mets	zip file structured according to the Metadata Encoding and Transmission Standard (METS)	No
mpeg-didl	a digital object made up of multiple digital items, described using the Digital Item Declaration Language (DIDL), part of MPEG-21	No
scorm	zip file structured according to the ADL SCORM specification (based on IMS Content Packaging). Possible versions include "1.2" and "2004".	Yes

The following example shows how the "package" extensions may appear in the context of a record returned as part of an SRU/SRW response. The "package" prefix is mapped to the XML namespace "info:srw/extension/13/package-v1.0".

```
<record>
  <recordSchema>info:srw/schema/1/dc-v1.1</recordSchema>
  <recordPacking>xml</recordPacking>
  <recordData>
```

```

<srw_dc:dc>
  <dc:title>This is a Sample Record</dc:title>
</srw_dc:dc>
</recordData>
<recordPosition>1</recordPosition>
<extraRecordData>
  <package:packageType>imscp</package:packageType>
  <package:packageTypeVersion>1.1.2</package:packageTypeVersion>
  <package:packagePreviewLocator>
    http://repository.mydomain.edu/open-preview?key=i175a12642t
  </package:packagePreviewLocator>
  <package:packageManifestLocator>
    http://repository.mydomain.edu/objects/121/files/imsmanifest.xml
  </package:packageManifestLocator>
  <package:packageDownloadLocator>
    http://repository.mydomain.edu/download-package?object-id=121
  </package:packageDownloadLocator>
</extraRecordData>
</record>

```

3.6. Metadata Formats

The metadata format returned by the SRU/SRW service can be controlled by setting the value of the "recordSchema" parameter in the SRU/SRW request. The following table gives details of the record formats currently supported in the SRU/SRW response.

Name	value	specification
Dublin Core	dc	http://dublincore.org/documents/dces
IEEE-LOM	lom	http://ltsc.ieee.org/wg12/materials.html

3.7. Access Control

The ability of external systems to search an intraLibrary repository is configurable for each collection within the repository. Two approaches are currently supported:

1. selectively opening/closing collections for external search
 - this is controlled by editing the properties of a collection in the "admin tools"

- interface
 - a collection is either open to external searching or closed to external searching (default)
- 2. controlling access using [SRU authentication tokens](#)
 - tokens are assigned to a collection by editing the properties of a collection in the "admin tools" interface
 - tokens can be supplied in the search request using the "x-authenticationToken" parameter
 - results will only be returned from collections to which the specified token has been assigned, or to which no tokens have been assigned

3.8. Modifying the Metadata Returned

The system can be configured to overwrite parts of the metadata returned in SRU/SRW responses. In the "General Configuration" section of the "system" page there is an option to select a metadata template to apply to search results. The values and macros in the selected template will overwrite the values stored in the relevant fields of each record returned in a search request.

Possible uses for the feature include:

- consistently exposing useful properties, such as PublicURLs in specific metadata fields
- blanking the contents of certain fields for privacy or security reasons
- ensuring consistency in important statements, such as those describing copyright or usage policy

3.9. Conformance

IntraLibrary complies with the [Base Profile Requirements for SRW](#). It conforms with "Level 1" of [CQL](#). Range queries are not currently supported.

IntraLibrary supports the DC context set, and returns metadata in Simple Dublin Core (default) and IEEE LOM formats.

4. File Transfer

Sometimes it is not practical to use the "upload" facility to put files and packages into intraLibrary. For example, a file/package may be too big to upload in a reasonable time over a restricted network connection, or there may be a large set of files that need to be transferred from an existing system. It is possible to provide other means for users to transfer files to the repository server, such as FTP, including Secure FTP (SFTP) and WebDav.

The "file import" action in an intraLibrary workflow allows users to import files from a specified folder on the server filesystem. The location of this folder is configurable by user group. Any FTP or WebDAV server can be used to give intraLibrary users controlled access to allow them to transfer files to the appropriate folder on the server. If you would like some more information about this, please contact Intrallect.

It is possible to automate the deposit of files/packages into the repository using the [Repository Java API](#). A sample implementation is provided with the product.

5. IMS DRI

The implementation of the IMS Digital Repository Interoperability (IMS DRI) "spec" in intraLibrary is based on a specification developed by the (ECL) project. This interface is deprecated, but still maintained for backward compatibility.

6. Repository Java-API

IntraLibrary has a fully documented "repository" Java-API. This API enables Java classes running in the same application context (webapp) as intraLibrary to access aspects of the intraLibrary business layer. It does not support Remote Method Invocation (RMI).

The Repository Java-API can be used to enable other software to:

- gather information about the status of a repository
- create and manage users
- trigger workflow events
- query the structure of a workflow and find learning objects within it
- search for objects

The following support information is provided with the product:

- complete set of JavaDocs
- some sample implementations, including full source code and installation instructions:
 - adding users to intraLibrary
 - uploading multiple files into intraLibrary
 - searching for learning objects in intraLibrary

7. External Authentication

In its default configuration IntraLibrary uses a built-in directory of user accounts when authenticating users. IntraLibrary can also be configured to query a chain of external sources of authentication when authenticating users. This means an intraLibrary repository may be used by an organisation that has one or more existing directories of users, or by a group of organisations working together, who each want to manage their user accounts separately.

An authentication chain is defined in a simple XML file in the configuration folder of the intraLibrary installation. When a user attempts to log-in to intraLibrary, their credentials are checked against every source of authentication in turn. A chain will typically include the internal directory of users in intraLibrary itself.

There are many possible kinds of external sources of authentication, including:

- a set of user accounts managed via the operating system of a local fileserver
- a database of users that is part of another enterprise application
- a directory exposed under the Lightweight Directory Access Protocol (LDAP) or Microsoft Active Directory
- a regional or national authentication service
- a federated authorisation mechanism, such as Shibboleth

IntraLibrary supports a pluggable authentication interface. An implementation of this authentication interface is required for each kind of authenticator used in an authentication chain. Full documentation of this interface is provided with the system, including the following reference implementations:

- a simple SQL-based authenticator
- an LDAP authenticator
- an integration with EduServ Athens Single-Sign On

8. Future Developments

Intrallect plans to implement the following interfaces in intraLibrary:

- Expose all the key data objects through the Java API, to facilitate the generation of custom reports
- Remote deposit into an intraLibrary repository using the service interface defined by the [JISC SWORD project](#). This will be based on the [Atom Publishing Protocol](#).
- Harvesting from other repositories via OAI-PMH.
- Searching other catalogues via SRU/SRW.
- Enable an intraLibrary repository to be searched via the FIRE interface.

Intrallect is considering the following interfaces for implementation in intraLibrary:

- Transfer content packages to/from intraLibrary using AICC's PENS specification
- Manage user accounts in intraLibrary using the IMS Enterprise specification
- Federated authorisation using Shibboleth
- Open URL

9. References

- Open Archive Initiative - Protocol for Metadata Harvesting (OAI-PMH): <http://www.openarchives.org/pmh/>
- OAI Repository Explorer: <http://re.cs.uct.ac.za/>
- Search and Retrieve URL (SRU): <http://www.loc.gov/standards/sru/sru-spec.html>
- Search and Retrieve Web-Service (SRW): <http://www.loc.gov/standards/sru/srw>
- Dublin Core Metadata: <http://dublincore.org/documents/dces>
- IMS Metadata: <http://www.msglobal.org/metadata/index.html>
- IEEE LOM: <http://ltsc.ieee.org/wg12/materials.html>
- Open Digital Rights Language (ODRL): <http://odrl.net/>
- IMS Digital Repositories Interoperability Specification: <http://www.msglobal.org/digitalrepositories>