



This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

A yellow rectangular box containing the Creative Commons Attribution-NonCommercial-NoDerivs 2.5 license summary. At the top is the Creative Commons logo (CC) and the text 'creative commons COMMONS DEED'. Below this is the license title 'Attribution-NonCommercial-NoDerivs 2.5'. The text is organized into sections: 'You are free:' followed by a bullet point 'to copy, distribute, display, and perform the work'; 'Under the following conditions:' followed by three icons and their descriptions: 'BY: Attribution. You must attribute the work in the manner specified by the author or licensor.'; a crossed-out dollar sign icon for 'Noncommercial. You may not use this work for commercial purposes.'; and an equals sign icon for 'No Derivative Works. You may not alter, transform, or build upon this work.' Below these are two more bullet points: 'For any reuse or distribution, you must make clear to others the license terms of this work.' and 'Any of these conditions can be waived if you get permission from the copyright holder.' At the bottom, it states 'Your fair use and other rights are in no way affected by the above.' and 'This is a human-readable summary of the [Legal Code \(the full license\)](#).' with a 'Disclaimer' link and a small icon.

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Integrating Journal Back Files into an Existing Electronic Environment

[Jason Cooper](#) describes how Loughborough University Library integrated a number of collections of journal back files into their existing electronic environment.

Introduction

When we purchased two collections of journal back files for hosting locally we knew that there would be some work involved in providing them to our patrons as a usable service. The key task we faced was to get our final solution neatly integrated into our existing electronic environment. We did not want our patrons to have to go to a stand-alone search page when they could use our federated search engine. We wanted our OpenURL resolver to be able to link patrons to the article whether they found it via our federated search engine or another resource that supports OpenURLs. We also wanted people to be able to link straight to an article or a journals index so that lecturers could add specific articles to their reading lists. Finally we knew that we needed to supply a browsable index for the whole of each collection of back files.

The Back Files to Be Integrated

The initial two collections of back files that were to be integrated were the Institution of Civil Engineers Virtual Library Archive (ICE) [1] and the Royal Society of Chemistry Journals Archive (RSC) [2].

The ICE collection contains the contents from 13 journals published between 1836 and 2001 and is approximately 20Gb in size. The RSC collection contains about 238,000 articles published between 1841 and 2004 and is approximately 143GB in size.

Both collections of back files contained the PDF copies of the articles and the metadata for each article. The metadata was stored in individual files in an XML format; the XML Schema was different for the two collections which added to complexity of using a single system to deliver the back files.

The Existing Electronic Environment

The electronic environment into which the collections of back files were to be integrated consisted of a number of different services. Our Library Management System (Aleph [3]), Federated Search Engine (MetaLib [4]) and OpenURL Resolver (SFX [5]) are all products of the same company (Ex Libris [6]) and have a reasonable level of integration between each of them.

The Reading list system (LORLS [7][8]) supports both traditional print materials and electronic materials and will generate OpenURLs for each item on a reading list where possible. It is also possible to add a specific URL to each item to give a direct link to the electronic resource/article.

It is also possible for lecturers to put links to resources on the University's virtual learning environment (Moodle [9]) for students taking their modules.

The Required Results

It was required that, from a patron's point of view, there would be nothing special about the back files and they would appear to be just another resource that the patrons could search in the same way as other databases. This would require each back file collection to be searchable from within MetaLib. The results shown to the patron should contain a direct link to the article itself where possible.

Our OpenURL resolver would be required to link to an article provided it was supplied with enough information, and failing that, it should link to the index system for the relevant back file collection to let the patrons browse to the article in which they are interested.

The back files should have an index system that would display for each collection of back files the list of journals covered; each being a link that would bring up a list of the articles in that journal that are contained in the back file. It was also a requirement to be able to deep-link into the index system so that each of the journals contained in a collection of back files could be catalogued separately in our Library Catalogue (which is part of Aleph).

The Chosen Method

The first decision that was made was how the system would present the individual files to the patron. It was decided that a CGI script would present the patron with a click-through licence with two links, one to accept the licence and one to reject it. If they click on the link that states that they agree to the licence the script would return the requested file. If they click on the other link it would redirect them to the Library's home page.

The second decision was how to enable MetaLib to search the contents of each collection. Our solution was to use Zebra [10] to provide a Z39.50 interface to the metadata for each collection. We already had experience of doing this to provide a Z39.50 interface to our Institutional Repository (DSpace[11]) and many of those scripts could be reused. These scripts gave us our common target for the format of the metadata (Dublin Core XML) as this would enable us to maintain just one set of scripts for converting to Marc and indexing with Zebra.

To enable SFX to direct patrons to the resource effectively, it would need a way to look up the URL for a specific article contained within a collection. To do this we decided upon creating a plug-in for SFX which would create a URL using the relevant ISSN, volume, issue and start page from the OpenURL and offer this as a link for journals in the collections. The CGI script would then work out the correct file ID for the article and redirect the browser to the click through licence CGI script with that file ID. If the CGI script failed to find a match for the provided data then the browser would be redirected to index for that back file collection.

The final decision was how to provide the browsable index for the individual back file collections. For this we decided to create a back-end database that would be used by a small set of CGI scripts to provide a browsable index. It was decided that the browsable index should not contain any search functionality as that is already provided by MetaLib via the Z39.50 interface.

Creating the Click-through Licence

The first task was to create the database that the script would use to look up the file to return for a specific ID. Each back file collection would have its own database consisting of a single table. This table consisted of two fields, the ID and the full path for the file with that ID. Inspecting the file names provided in the two back file collections showed them to be unique within their own collection and so were adopted as a simple ID that would be contained within the metadata,

thus avoiding complexities for later stages of the development.

We created a database for each collection and populated it using a Perl script that parsed the tree structure of the collection and stored the full path for each file against the ID.

Once the database for each collection was populated, a Perl CGI script was created that accepts an ID. It then displays a licence to the patron with the accept and reject options (see Figure 1). The option to accept the licence links to the same CGI script with the ID parameter and an extra parameter stating that they had accepted the licence. The option to reject the licence is a link out of the system.

If the CGI script is called with the ID parameter and the parameter stating the patron has accepted the licence, then the script uses the ID to look up the path to the file in the database and it then returns that file to the patron.

The directory in which this CGI script is located on the server is password-protected; this measure limits access to the actual files down to members of the University.

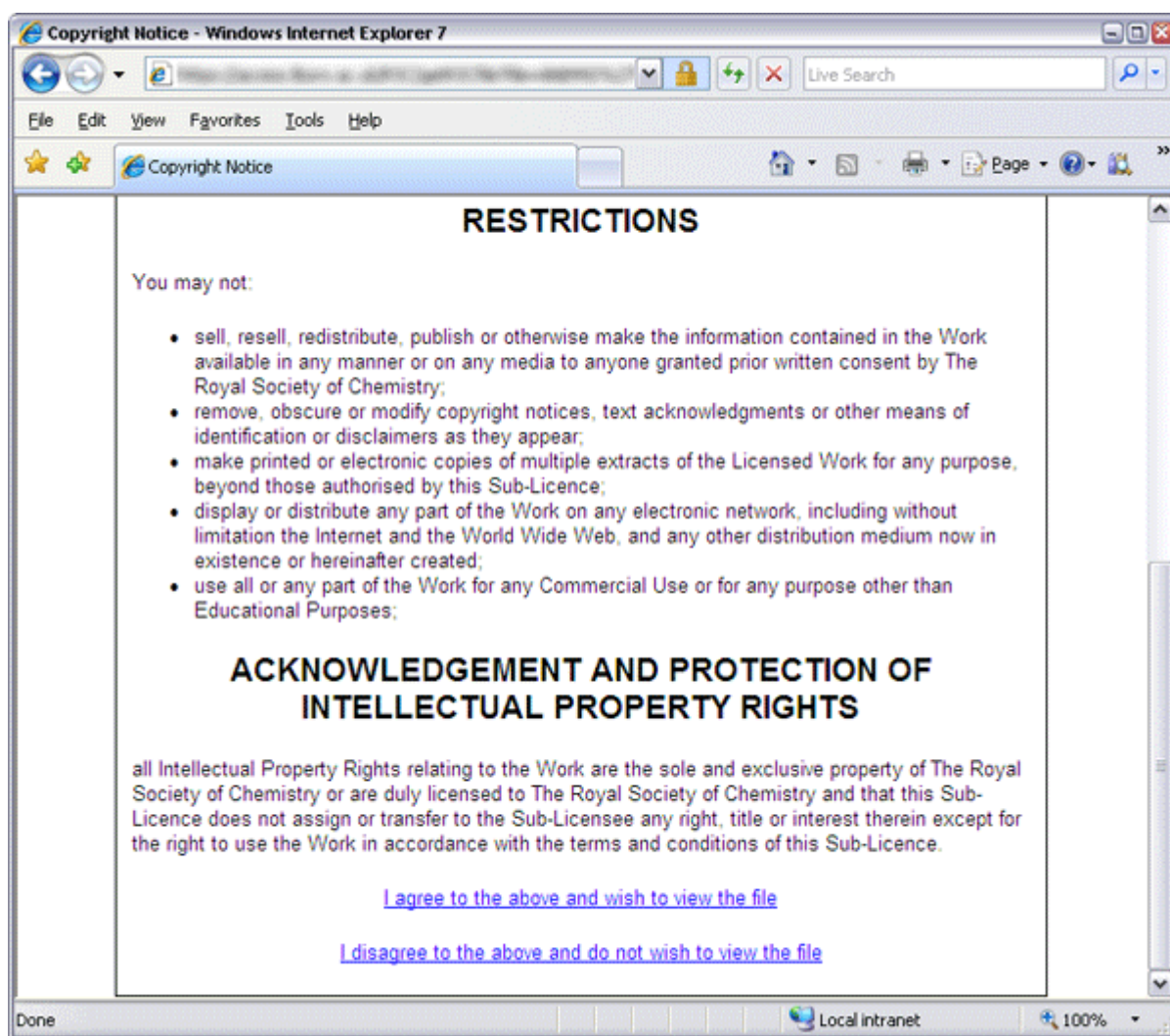


Figure 1: Screenshot of lower half of click-through licence

The Creation of the Z39.50 Interface

As we already had scripts for converting Dublin Core XML files into Marc for easy indexing in Zebra, we only had to concentrate on the scripts to map each collection's metadata into Dublin Core XML. This mapping was done via a Perl script that was custom-tailored to each collection's metadata. The script took a path to the root of the metadata files as a parameter

and then it would parse the directory tree under that and convert all the metadata into a single file in a Dublin Core XML format. Each record in the resulting Dublin Core XML contained the following metadata if it was available. In the case of the URL element the conversion script would generate this based upon the file name of the article (which was being used as a unique ID) and the known URL of the click-through licence CGI script.

- Article title
- Publisher
- URL that would link directly to the click-through licence CGI script with the ID for that article
- Authors
- Journal title
- Journal Date
- Journal Volume
- Journal Issue
- Article Description (The first 1024 characters of the article)

Our existing scripts were then used to convert the Dublin Core XML files into USMARC files which were then indexed by Zebra. The Z39.50 interface was tested using the yaz-client tool from the yaz tool kit [12]. At this point we then had a Z39.50 interface to search our back file collections.

Integrating the Back Files with MetaLib

After the Z39.50 interface was set up, the next step was to configure MetaLib to be able to use it as a resource. As we had reused our scripts from our institutional repository set-up and we already had a Z39.50 interface to our institutional repository, we could just copy the MetaLib configuration for that. The only changes needed were to point MetaLib at the correct Z39.50 interface and set the correct database within that to be searched, as each back files collection had its own database within Zebra.

Integrating the Back Files with SFX

To successfully integrate our back files into our electronic environment we needed to make sure that SFX would be able to locate articles within it from the usual information passed in an OpenURL. For each collection we created a database that contained for each article the file ID, journal ISSN, journal volume, journal issue and start page. We created a CGI script that uses this database to look up the file ID based on passed details and then redirects the patron's browser to the relevant file. If the CGI script does not receive enough information to identify the file uniquely, then it redirects the patron's browser to the browsable index for the relevant collection.

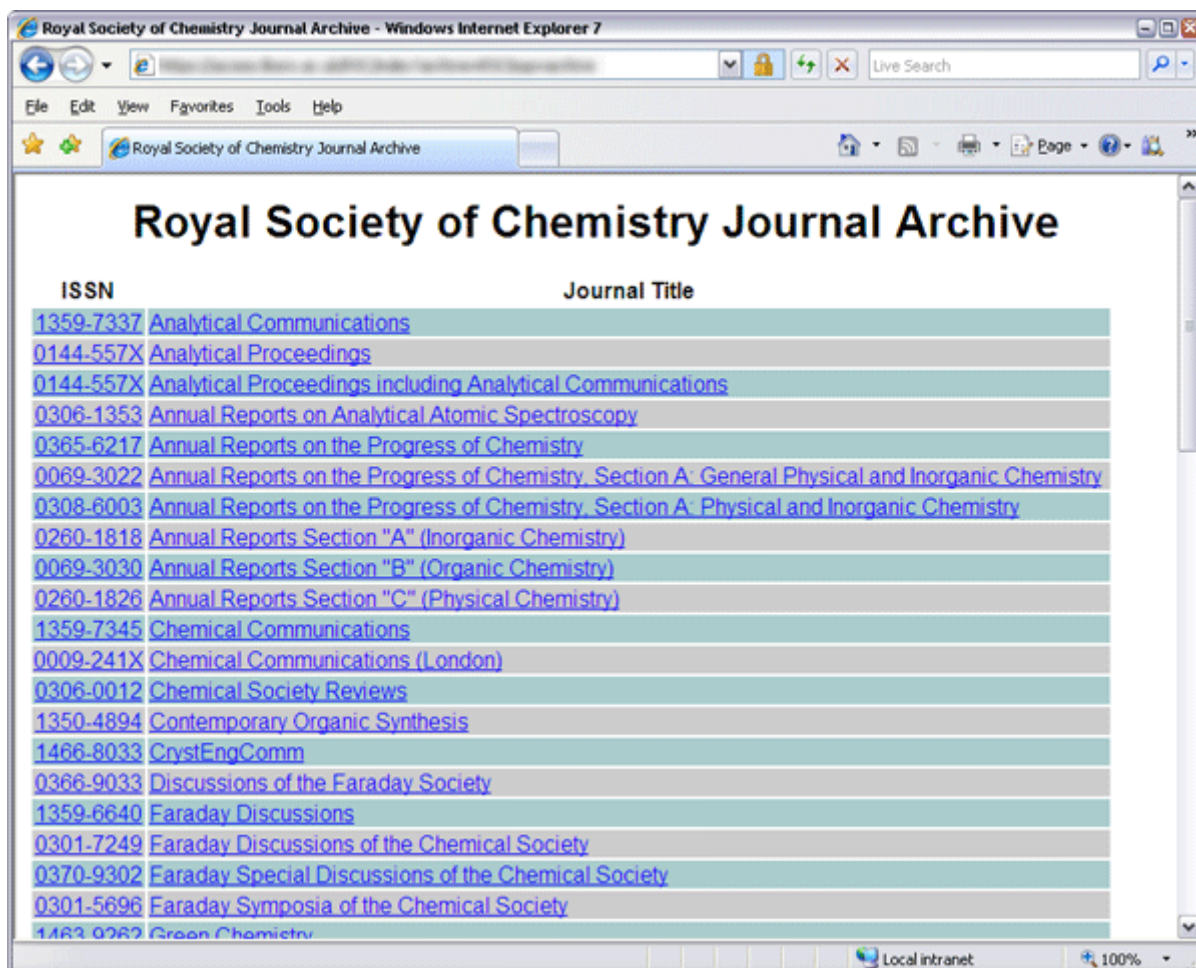
We then created an SFX plug-in that could be used by each of the journals in the archives. First the plug-in checks if the OpenURL passed to it contains a direct link to the required article (such as the results from a MetaLib search would produce). If it does, and the URL matches the pattern for our server, then it offers the patron that as one of the options. If it does not, then the plug-in offers a link to the above CGI script which it generates using the information passed to it in the OpenURL. This has the advantage of reducing look-ups on remote servers when SFX is processing an OpenURL.

Creating a Browsable Index

To create the browsable index of the back file collections, another database was created that would store some of the metadata for each article and to which journal they relate. It also stores

the archive to which the journal belongs. This was then populated via a Perl script from the processed metadata used by the Z39.50 server.

A set of Perl CGI scripts was created to provide the browsable index. To make it easy to deep-link into specific parts of the index, a Representational state transfer (REST) interface was used. In the browsable index there are three different levels available to patrons: the first is a list of the archives available which is never really used as most resources link through to one of the next two levels, but it is there for completeness; the second level is a list of all journals in the archive specified in the URL (see Figure 2) - this is the level that is linked to by most resources when referencing one of our archives; the third level is a list of all articles in a journal and is linked to mainly from our Library catalogue.



ISSN	Journal Title
1359-7337	Analytical Communications
0144-557X	Analytical Proceedings
0144-557X	Analytical Proceedings including Analytical Communications
0306-1353	Annual Reports on Analytical Atomic Spectroscopy
0385-6217	Annual Reports on the Progress of Chemistry
0069-3022	Annual Reports on the Progress of Chemistry Section A General Physical and Inorganic Chemistry
0308-6003	Annual Reports on the Progress of Chemistry Section A Physical and Inorganic Chemistry
0260-1818	Annual Reports Section "A" (Inorganic Chemistry)
0069-3030	Annual Reports Section "B" (Organic Chemistry)
0260-1826	Annual Reports Section "C" (Physical Chemistry)
1359-7345	Chemical Communications
0009-241X	Chemical Communications (London)
0306-0012	Chemical Society Reviews
1350-4894	Contemporary Organic Synthesis
1466-8033	CrystEngComm
0386-9033	Discussions of the Faraday Society
1359-6640	Faraday Discussions
0301-7249	Faraday Discussions of the Chemical Society
0370-9302	Faraday Special Discussions of the Chemical Society
0301-5696	Faraday Symposia of the Chemical Society
1463-0262	Green Chemistry

Figure 2: Screen shot of the browsable index

Conclusion

By integrating our back files with the rest of our electronic environment, rather than developing a stand-alone system, we cut out a lot of development work that would have been required. There was no need to develop a custom search interface to the back files as our patrons use MetaLib which has many more features than we could have developed in the time available to us. At the same time we were able to make it easier for our patrons to access the back files, through searching them via MetaLib and through SFX being able to point patrons to the articles in the back file collections from other resources that support OpenURLs.

References

1. JISC Collections : Institution of Civil Engineers Virtual Library Archive http://www.jisc-collections.ac.uk/catalogue/coll_icevirtuallib.aspx
2. JISC Collections : RSC Journals Archive http://www.jisc-collections.ac.uk/catalogue/rsc_journals_archive.aspx
3. Aleph Integrated Library System <http://www.exlibrisgroup.com/category/Aleph>
4. MetaLib: Reach Out and Discover Remote Resources <http://www.exlibrisgroup.com/category/MetaLibOverview>
5. SFX: Overview <http://www.exlibrisgroup.com/category/SFXOverview>
6. Ex Libris <http://www.exlibrisgroup.com/category/Home>
7. Loughborough Online Reading List System (LORLS) <http://lorls.lboro.ac.uk/about.html>
8. Brewerton, G. and Knight, J., 2003. From local project to open source: a brief history of the Loughborough Online Reading List System (LORLS). VINE, 33(4), pp. 189-195 <http://hdl.handle.net/2134/441>
9. Moodle - A Free, Open Source Course Management System for Online Learning <http://moodle.org/>
10. Zebra <http://www.indexdata.dk/zebra/>
11. dspace.org <http://www.dspace.org/>
12. YAZ <http://www.indexdata.dk/yaz/>

Author Details

Dr. Jason Cooper

Library Systems Analyst\Programmer
Loughborough University

Email: j.l.cooper@lboro.ac.uk

Web site: <http://www.lboro.ac.uk/library>

Article Title: "Integrating Journal Back Files into an Existing Electronic Environment"

Author: Jason Cooper

Publication Date: 30-July-2008 Publication: Ariadne Issue 56

Originating URL: <http://www.ariadne.ac.uk/issue56/cooper/>

[Copyright and citation information](#)