Title: Optimizing Metadata in Metalib for Quality OpenURLs

Summary: This presentation is about tweaking settings in the "Conversion" and "OpenURL" tabs of the Metalib IRD configuration in order to generate quality OpenURLs that will more reliably lead to full-text articles.

INTRODUCTION

We all know it is an imperfect world, and what could be more imperfect in this imperfect world than bibliographic data, especially when it comes from a variety of sources? While every citation received by MetaLib may have a title in the 245 field, that may be all there is in common among MARC records received from different vendors.

In MetaLib Ex Libris has given us a number of tools which can be used to reconfigure bibliographic data so that it will display properly and link properly. These functions can be manipulated via the Conversion and OpenURL tabs in the configuration record and by the native interface fields on the Subscription and Primary Presentation tabs of the resource records. To be sure Ex Libris has usually provided some conversion already, but the experience at our institution has been that some of these conversions may be incomplete or out of date. Consequently we have found ourselves reviewing the configuration and resource records of databases that are important to our users in order to ensure optimal performance.

In this presentation we are going demonstrate the use of the MetaLib tools through various cases that we have encountered during the past couple of years. We hope that they will give you some sense of the power contained within them to address a multiplicty of issues relating to the presentation and linking capability of metadata.

Before we look at specific cases, we want to familiarize you with the features in Metalib that we are targeting(no pun intended) so that as we run through the cases you won't find yourself lost thinking, where in the heck is that screen, or what did he just do.

The Metalib IRD consists of two main parts—the part that contains institutional subscription information, and the part that contains the configuration for actually talking to the resource at the other end.

The configuration part is the part that we are interested in. This is the part that understands not only how to ask questions of the database, but it understands the structure of the record that is coming back from the database. Ex Libris has already done a significant amount of configuration to make these work, so what we are talking about is fine tuning. The fine-tuning is not trivial as you will see in that it often makes the difference between getting to the full-text of an article or just to a journal level page. This is not where we want our users to find themselves.

The configuration tab has to do with locating and relocating the metadata in expected, and not expected locations. The OpenURL tab has to do with recognizing the metadata and putting it properly and successfully into an OpenURL

Now let me just mention that there is a dark side to this—dark for some, fun for others—regular expressions may be a useful/necessary evil in this game. It's not the only tool in the bag, but it may be the sharpest and finest for certain tasks we need to accomplish.

CASE 1 Citation Data - Volume and Issue Numbers in General OneFile

<1.1**>** Let's start off with a simple but trenchant example of how the Parser on the OpenURL tab can be used to correct a metadata inconsistency in a database. The database is Gale General OneFile. The field we will observe is the 773 field, which according to the MARC standard holds the host citation data, that is, the journal title, the date, the volume number, the issue number, and the page numbers.

Note that the citation line in the example provides complete metadata which should theoretically generate a complete OpenURL. **<**1.2**>** However, when we look at the top of the SFX Services Menu there are signs of a problem. Neither the volume nor the issue number appears and consequently, **<**1.3**>** when we click on the JSTOR link, we are taken to the homepage of the journal, a typical sign of incomplete metadata in an OpenURL. **<**1.4**>** If we look at the OpenURL, we do indeed see that the volume number and the issue number are lacking.

Why is this? After some analysis we found out that Gale had changed the format of the citation information in both Academic OneFile and General OneFile in about 1999. **<**1.5**>** Prior to 1999 the metadata in the 773 $g

was structured like this: after the year there is a comma and a space, then the letter v and the volume number, another comma and a space, then the letter n and the issue number, another comma and a space, then the letter p and the page number or numbers. **<**1.6**>** From about 1999 on the subfield g was restructured with no comma after the volume number and the letter i instead of the letter n preceding the issue number. **<**1.7**>** When the hard-working people at Ex Libris put together the entries on the OpenURL tab of the configuration records for Academic OneFile and General OneFile, they were not aware of the change in format and so only accounted for the later format of the subfield. Consequently all the citations in each database for the period prior to 1999 were not resolving to the article level through SFX.

What to do? **<**1.8**>** We corrected this by creating a local version of the configuration files and customized them to account for the different format of the volume and issue numbers. Here is the revised OpenURL tab with additional entries for the volume number to account for the presence of a comma and for the issue number to account for the presence of the n instead of the i. Both of these use the Parser, the use of which is described in the Resource Management Guide, section 8. What the syntax for the issue number says is to take field 773, subfield g, and starting with a space-n, go all the way to the comma.

The result? **<**1.9**>** The SFX Services Menu for the citation now displays the volume and issue number, meaning that the OpenURL contains them. **<**1.10**>** On clicking the JSTOR option we are taken to fulltext at the article level. By this means we have restored article level resolution to the entire database. We also reported our findings to Ex Libris so that the configuration records for the databases could be updated in the Central KnowledgeBase.

<** **> Questions???

CASE 2 Citation Data - Merging Multiple Fields into One 773 for Display Purposes

<** **> Let's look at a case of improving the bibliographic display. **<**2.1**>** Here is a citation from the Directory of Open Access Journals. Pay particular attention to the citation information. It is awkward having the data on separate lines and even more awkward that the elements are not

labeled. <**2.2**> Looking at the MARC record, one can see that pages are present, <**2.3**> but not represented in the public display. I don't think that our users are very likely to refer to the MARC record to figure out which field is which.

<**2.4**> Looking at the configuration record, we can see that an attempt has been made to merge the 773 subfields t, v, m, and z into one 773 with repeating subfields g. However, it didn't work because the wrong Action was used. Merging multiple fields into a single field with repeating subfields requires the Merge_sub action.

<**2.5**> And that is what we used when we modified the configuration record. We started a new 773 field with the title in $t. That uses the Merge action. <**2.6**> Then we created repeating subfields g for the year, the volume, the issue, and the pages with the Merge_sub action. In addition, we used Regular Expressions to insert a label in front of the data in the subfields g.

<**2.7**> The result was a nicely formatted citation information field that makes clear exactly what type of information each subfield contains.
<**2.8*> In the converted MARC record you can see how the subfields are arranged. Unfortunately the Merge_sub action does not enable one to merge multiple fields into the same subfield, but this is at a satisfactory substitute.

<** **> Questions???

CASE 3 DOI - Positioning it for Display Purposes and in OpenURL

<** **> Let's move on to a case that involves both the bibliographic display and the OpenURL. The DOI or Digital Object Identifier is a good example. As you may know, it can be used in an OpenURL to access citation and fulltext linking information stored in the CrossRef registry. The interaction between SFX and CrossRef can be used to point to the appropriate copy.

After we registered with CrossRef so that we could use SFX to access their database, we cycled through our databases looking for those which contained DOIs so that we could test them. What we found was that DOIs came in such a variety of fields that they were often hidden from the bibliographic display and they were seldom activated for OpenURL

generation. Consequently we set about rectifying the situation. In the end we modified the configuration records for 19 databases so that we could either display the DOI or use it in an OpenURL. Let's look at an example.

<**3.1**> MetaLib expects a DOI to be in the 024 field of the MARC record. <**3.2*> If that field is present, it will display with an "IDNumber:" label. <**3.3*> We didn't even know that EBSCO's Business Source Elite contained DOIs until we started snooping in the MARC record because they come in a 028 $a, not a 024. <**3.4**> Hence they were not appearing in the bibliographic display. <**3.5*> When we discovered this, we made a change to the Conversion tab <**3.6**> that placed the DOI in the 024 field of the converted MARC record. <**3.7**> Consequently, it appears with the "IDNumber:" label in the bibliographic display .

However, there was another problem. <**3.8**> Business Source Elite records already contain 024 fields with NAICS Industry Codes in them. If we generated the DOI OpenURL element from the 024 field, it was likely that the wrong number would be placed in it.

<**3.9**> Taking a leaf from the model provided by Ex Libris for the PUBMED database, we created a second field from the 028 called DOI. <**3.10**> Then on the OpenURL tab of the configuration record, we assigned the DOI field to the OpenURL Element DOI. <**3.11**> That resulted in the DOI being inserted into the appropriate field of the OpenURL, <**3.12**> thus leading to the fulltext of the appropriate copy for the library's users.

<** **> Questions???

CASE 4 Control numbers – Link to Native Interface (001, 035)

<** **> Let's look at a slightly different case of linking: linking back to the native interface. As useful as MetaLib is, users sometimes need to look at full records in the native interface, perhaps because they contain information that MetaLib does not receive from the source, perhaps because the user wants to look at information in a familiar format.

<**4.1**> Here is a record from the University of Chicago's catalog. <**4.2**> Look carefully at the holdings view. Is the item really on the shelf and available for checkout or not? I don't know about you, but I have a

hard time interpreting the holdings information. <**4.3**> Clicking on the Resource link at the top of the MetaLib record takes me to the native interface where it unambiguously states that the item is not checked out.

How did we make this type of link work? <**4.4**> First we looked at the MARC record for some kind of control or accession number. Catalogs vary, but typically such numbers are in the 001 field or the 035 field. The University of Chicago places control numbers in the 035 field. Then we searched for some kind of linking syntax that allows the insertion of the control number into an HTML string. Sometimes you can deduce the linking syntax by performing searches in the native interface and observing the formation of the URLs. At other times you need to contact the vendor of the software or the library that runs it.

<**4.5**> Once the URL syntax was determined, we entered a template URL into the Link to Records in Native Interface on the Subscription tab of the Resource record. <**4.6**> Let's look at the template URL closely. It contains the host name and a series of parameters that point to the correct index and display profile. The control number itself is represented by a dollar sign, the MARC field number, an underscore, and a subfield. When a user clicks on the Resource link of the MetaLib screen, the value of the first 035 $a is substituted in the template and the complete URL is transmitted over the Internet.

<**4.7**> Here is another example. We happened to notice that records in MetaLib do not include all the fields displayed in the native interface of Biological Abstracts as supplied by ISI. <**4.8**> Consequently we setup a link to the native interface via the accession number in the AN field. <**4.9**> The linking was accomplished by taking the accession number from the AN field and plugging it into a template URL in the MetaLib Resource record. <**4.10**> The native record for this example contains a list of chemical names and methods not included in the MetaLib record as well as a link to another article that cited this one. Such citing information is extremely important to researchers.

<** **> Questions???

Issues Conclusions

I don't think that local customizations are avoidable if one really wants to

provide top level service to one's user community. Bibliographic data is so imperfect and vendors change its format so often that Ex Libris simply cannot keep up with it. That is why they provided the toolkits on the Conversion and OpenURL tabs of the config record.

What is the appropriate level at which to resolve metadata issues? Ideally, they should be resolved as close as possible to the source of the data, preferably upstream from the library and its users. It may be the publisher or society that generates the data; it may be the vendor that makes it available to potential users. However, resolution may devolve onto the federated search software such as MetaLib, the library which uses the tools provided by the federated search software, or an end user interface such as Xerxes.