The aim of this presentation is to highlight some easy enhancements which we can all make to our Aleph OPACs. Some of these suggestions are well-known, but some less so.
Missing links

- Link to full record from search results

First we look at “missing links”. There are several places in the Aleph OPAC where it would be helpful to have links and where none are present. In many cases we can add them easily.
The default Aleph OPAC has a small, easily missed, link to the full record display. We want a link from the title.
This is all we need. You may need to adjust the placeholder numbers if your title display is not in column 5 of the short results table. The JavaScript takes the placeholder for column 1, the number with link, and extracts the URL from it. This is then wrapped round the title. It is all done in such a way that if the user has no JavaScript support the title will still appear.
And here is the result! Much better. This enhancement is very well-known, and was in fact recommended to us by someone from Ex Libris. When will the OPAC be enhanced to avoid having to use JavaScript for this purpose?
Missing links

- Link to full record from search results
- Links in My Library Card to expanded information

There is a similar situation in My Library Card.
Loan history list for: PHILLIPS, MATTHEW E (Panini20)

63 loans out of a total of 63 loans are listed.

For more details about a loan, click on the underlined number.

<table>
<thead>
<tr>
<th>No.</th>
<th>Author</th>
<th>Title</th>
<th>Year</th>
<th>Due date</th>
<th>Due hour</th>
<th>Returned date</th>
<th>Return hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Palmer, Martin</td>
<td>Making the most of RFID in libraries</td>
<td>2009</td>
<td>25/08/2009 23:59</td>
<td>07/08/2009 16:52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Manning, Olivia</td>
<td>The Balkan trilogy / Works, Complete edition</td>
<td>1908</td>
<td>24/06/2009 23:59</td>
<td>08/06/2009 08:57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Lawrence, D. H. (David Herbert), 1865–1930</td>
<td>Women in love / Works, Complete edition</td>
<td>1903</td>
<td>29/05/2009 23:59</td>
<td>20/05/2009 16:59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Wall, Larry</td>
<td>Programming Perl.</td>
<td>2000</td>
<td>06/05/2009 23:59</td>
<td>07/05/2009 10:05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Small links from the numbers to the expanded information.
<table>
<thead>
<tr>
<th>Loan history list for: PHILLIPS, MATTHEW E (Panini20)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bor-history-loan-body</strong></td>
</tr>
<tr>
<td>Replace <code>&lt;td class=td1 valign=top&gt;$0200&lt;/td&gt;</code> with</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><code>&lt;td class=td1 valign=top&gt;</code></td>
</tr>
<tr>
<td><code>&lt;script language=&quot;javascript&quot;&gt;</code></td>
</tr>
<tr>
<td>var recordLink=&quot;$0100&quot;;</td>
</tr>
<tr>
<td>recordLink = recordLink.substring(0, recordLink.indexOf(&quot;&quot;&gt;&quot;));</td>
</tr>
<tr>
<td>document.write(recordLink);</td>
</tr>
<tr>
<td>document.write(&quot;&quot;&gt;&quot;);</td>
</tr>
<tr>
<td>&lt;/script&gt;$0200</td>
</tr>
<tr>
<td><code>&lt;script language=&quot;javascript&quot;&gt;</code></td>
</tr>
<tr>
<td>document.write(&quot;&lt;/a&gt;&quot;);</td>
</tr>
<tr>
<td>&lt;/script&gt;</td>
</tr>
<tr>
<td>&lt;/td&gt;</td>
</tr>
</tbody>
</table>

A very similar piece of JavaScript.
And here is the result. It is much clearer to the user that there are links to more information. There are several other screens in My Library Card which can benefit the same way.
Missing links

- Link to full record from search results
- Links in My Library Card to expanded information
- Links back to catalogue from expanded screens
This is what our expanded information for a loan history item looks like. We have added a link (via the barcode index) to take you back to a catalogue search for the item you borrowed. This is handy if the user is wanting to place a hold request for something he/she had borrowed before.
Loaned Item Details

<a href="&server_f?func=findb&request=$1700&local_base=DUN01&find_code=BAR">View item in catalogue</a>

This time no Javascript is required: even better!
And the link takes you through to the catalogue search results.
Missing links

- Link to full record from search results
- Links in My Library Card to expanded information
- Links back to catalogue from expanded screens
- RSS links for SDI
This is the screen in My Library Card showing the users alerts or SDI requests. The RSS URLs are not clickable, limiting their usefulness.
Replace

\(<td\ class=td1>\$0800</td>\)

with

\(<td\ class=td1><a\ href="\$0800">\$0800</a></td>\)

Again, we can fix this without using Javascript.
Now the links are clickable.
Missing links

- Link to full record from search results
- Links in My Library Card to expanded information
- Links back to catalogue from expanded screens
- RSS links for SDI
- Remember to test with JavaScript disabled

When making enhancements it is very important to test the functionality with JavaScript disabled. This is because some users do not have access to JavaScript-enabled browsers, or prefer to disable JavaScript for security or privacy reasons. The main thing to ensure is that your “enhancement” does not reduce the functionality for these users. Sadly I have seen examples of Aleph OPACs where the use of JavaScript for quite minor purposes (e.g. preventing users double-clicking in error) has made parts of the interface unusable for those who do not have JavaScript available.
Social bookmarking

- Creating a permanent URL when the result has context within a set of results

http://library.dundee.ac.uk/F/
5DIT2RYUTA2RA8YE36D5MQS622ENTA842R6
HB1CX14S5NXFBB5-36854
?func=full-set-set
&set_number=195186
&set_entry=000002
&format=999

This is a typical URL from a full record display in Aleph. If the user bookmarks the page the bookmark will stop working once VIR01 is cleared, because the URL depends on a session ID and a search set.
Social bookmarking

- Creating a permanent URL when the result has context within a set of results
- AddThis service: http://www.addthis.com/

We have added a bookmark button via the free “AddThis” service.
Here’s what it looks like in our OPAC.
Several different services are available on the pop-up menu.
Even more are available on the full menu. All of this is maintained by AddThis so we do not have to work out the API for each individual service.
Here we are in Delicious, bookmarking one of our records.
Important note

Please note that the following slides show a method which is now out of date. Please instead refer to the improved method at

http://www.exlibrisgroup.org/display/Aleph/AddThis+bookmarking+from+Aleph+full+record
**full-set-head-***, **direct-head-***
(There may be several files to alter, e.g. full-set-head-bor, full-set-head-ill, etc.)

HTML to make AddThis button appear in the button bar:

```html
<td class="bar" align="center" nowrap>
  <a href="http://www.addthis.com/bookmark.php"
      onmouseover="return addthis_open(this, ",",
      dunBookmark,
      dunTitle)"
      onmouseout="addthis_close()"
      onclick="return addthis_sendto()">
    <img src="http://s9.addthis.com/button1-bm.gif" width="125"
      height="16" border="0" alt=""/>
  </a>
</td>
```

Note that these instructions have been superseded by the improved version at http://www.exlibrisgroup.org/display/Aleph/AddThis+bookmarking+from+Aleph+full+record
full-set-head-***, direct-head-***
Just above bottom line, above the <table> tag:

<!-- START: AddThis.com button -->
<script type="text/javascript">
    addthis_pub = 'addthisusername';
    addthis_logo = 'URL of banner graphic for full window';
    addthis_logo_background = '0F228B';
    addthis_logo_color = 'FFFFFF';
    addthis_brand = "";
</script>
<script type="text/javascript" src="http://s7.addthis.com/js/152/addthis_widget.js"></script>

continued on next slide…

Note that these instructions have been superseded by the improved version at http://www.exlibrisgroup.org/display/Aleph/AddThis+bookmarking+from+Aleph+full+record
... continued:

```javascript
<script type="text/javascript">
  var dunBookmark="&server_httpd";
  dunBookmark +=
    "/F/?func=direct&local_base=$0900&doc_number=$0600&dunsid=addthis";
  dunTitle = "Title";
</script>
<!--[END: AddThis.com button]-->
```

**direct-head-***
Similar, but use
func=direct&local_base=$0300&doc_number=$0100

Note that these instructions have been superseded by the improved version at
http://www.exlibrisgroup.org/display/Aleph/AddThis+bookmarking+from+Aleph+full+record
... continued:

```javascript
<scripxt type="text/javascript">
    var dunBookmark = "&server_httpd;"
    dunBookmark +=
        
```

• Appears as the title of the web page in the
  bookmarking service.
• Hard to get the actual book title to appear.

```javascript
    func=direct&local_base=$0900&doc_number=$0600&dunsid=addthis";
    dunTitle = "Title",
</script>
<!-- END: AddThis.com button
</script>
```

**direct-head-***
Similar, but use
func=direct&local_base=$0300&doc_number=$0100

Note that these instructions have been superseded by the improved version at
http://www.exlibrisgroup.org/display/Aleph/AddThis+bookmarking+from+Aleph+full+record

The improved version obtains the title from the page and passes it on to the bookmarking
service.
The “dunsid” parameter is ignored by Aleph but allows us to track usage of the facility via our Apache logs.

Note that these instructions have been superseded by the improved version at http://www.exlibrisgroup.org/display/Aleph/AddThis+bookmarking+from+Aleph+full+record
The usage of links bookmarked via this service at Dundee. Not really very well used. Sadly you cannot really predict which features are going to be popular until you have released them!
Embedding search boxes

- MetaLib deep linking guide

It is easy to embed search boxes for MetaLib in other web pages. Consult the Deep Linking Guide which forms part of the MetaLib documentation.
We have added it to our Library home page (see the boxes at the bottom). We have an embedded box for the catalogue also.
Embedding search boxes

- Code for an Aleph search box:

  `<form method="get" action="http://library.dundee.ac.uk/F/">
  `<input type="hidden" name="func" value="find-e" />
  `<input type="hidden" name="adjacent" value="n" />
  `<input type="hidden" name="find_scan_code" value="FIND_WRD" />
  `<input type="hidden" name="dunsid" value="libweb" />
  `<input type="submit" value="Search Catalogue" />
  </form>`

This is all you need for an embedded catalogue search box.
Embedding search boxes

- Code for an Aleph search box:

```html
<form method="get" action="http://library.dundee.ac.uk/F/">
<input type="hidden" name="func" value="find-e" />
<input type="hidden" name="adjacent" value="n" />
<input type="text" name="request" value="" size="13" />
<input type="hidden" name="find_scan_code" value="FIND_WRD" />
<input type="hidden" name="dunsid" value="libweb" />
<input type="submit" value="Search Catalogue" />
</form>
```

We’re using find-e: you could use find-b

You may wish to change the search method here.
Embedding search boxes

- Code for an Aleph search box:

  ```html
  <form method="get" action="http://library.dundee.ac.uk/F/">
    <input type="hidden" name="func" value="find-e" />
    <input type="hidden" name="adjacent" value="n" />
    <input type="text" name="request" value="" size="13" />
    <input type="hidden" name="find_scan_code" value="FIND_WRD" />
    <input type="hidden" name="dunsid" value="libweb" />
    <input type="submit" value="Search Catalogue" />
  </form>
  ```

  Defines which index to search. Or you could have a `<select>` drop-down menu of indexes.

If you use “find-b” the indexes are expressed differently.
Embedding search boxes

- Code for an Aleph search box:

  ```html
  <form method="get" action="http://library.dundee.ac.uk/F/">
  <input type="hidden" name="func" value="find-e" />
  <input type="hidden" name="adjacent" value="n" />
  <input type="text" name="request" value="" size="13" />
  <input type="hidden" name="find_scan_code" value="FIND_WRD" />
  <input type="hidden" name="dunsid" value="libweb" />
  <input type="submit" value="Search Catalogue" />
  </form>
  
  I use the “dunsid” hidden field to track usage of the feature via the Apache logs
  
  Again, we use a “dunsid” parameter which Aleph ignores to track the usage of the feature via the Apache logs.
This is the usage of the embedded catalogue search box on our home page. It is very popular. Notice how the usage increased markedly at the start of the new academic year in September 2008. This is clearly because the habits of our new users are different. People who have been at the university longer take a while to change their habits and how they access the catalogue.
Embedding search boxes

You can export quickset details and links to resources from MetaLib and include in dynamic web pages to save maintenance effort.

Although this might look like a straightforward embedded MetaLib search box, the page is written in PHP and pulls the quickset information (the names of the databases and the links to their native interfaces) direct from our MetaLib server on the fly. This is not an easy enhancement, but it saves a lot of work maintaining web pages.
CrossSearch is our branding for MetaLib. Here is an example of a MetaLib quickest being available on the browser search menu.
Browser search plugins
Browser search plugins

QuickSearch

Search for "perf" in "Library catalogues"

<table>
<thead>
<tr>
<th>Database Name</th>
<th>Status</th>
<th>View retrieved</th>
<th>Cancel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Andrew's Library Catalogue</td>
<td>DONE</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>University of Alberta Dental Library Catalogue</td>
<td>DONE</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>University of Dundee Library Catalogue</td>
<td>DONE</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

* Your search is taking place, and results will be found and then retrieved in batches of up to 50 results from each database. When the search is complete, a list of results will be displayed.
Browser search plugins

- First create XML describing service:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-spec/opensearch/1.1/">
  <ShortName>UoD Library catalogue</ShortName>
  <LongName>University of Dundee Library catalogue</LongName>
  <Description>Performs a keyword search on the University of Dundee Library catalogue</Description>
  <InputEncoding>UTF-8</InputEncoding>
  <Tags>university dundee library catalogue</Tags>
  <Developer>IT Support, University Library, Dundee</Developer>
</OpenSearchDescription>
```

continued …

To create a browser search plugin, you first need an XML file describing the search service. Here is the first half.
Browser search plugins

- First create XML describing service:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
  <ShortName>UoD Library catalogue</ShortName>
  <LongName>University of Dundee Library catalogue</LongName>
  <Description>Performs a keyword search on the University of Dundee Library catalogue </Description>
  <InputEncoding>UTF-8</InputEncoding>
  <Tags>university dundee library catalogue</Tags>
  <Developer>IT Support, University Library, Dundee</Developer>
</OpenSearchDescription>
```

Descriptive stuff for humans. ShortName limited to 16 characters.

The first half of the description is just stuff for humans. Adjust as required for your search service.
Browser search plugins

… continued

<Url type="text/html"
    template="http://library.dundee.ac.uk/F/?func=find-
    e&amp;adjacent=N&amp;request={searchTerms}&amp;find_s
    can_code=FIND_WRD&amp;dunsid=IE7search"/>
<Image height="16" width="16" type="image/x-
    icon">http://www.dundee.ac.uk/library/graphics/dundee-shield-
    16.ico</Image>
</OpenSearchDescription>

The second half of the file describes the service to the browser.
Browser search plugins

… continued

<Url type="text/html"
    template="http://library.dundee.ac.uk/F/?func=find-
    e&adjacent=N&request={searchTerms}&find_s
    can_code=FIND_WRD&dunsid=IE7search"/>

<Image height="16" width="16" type="image/x-
    icon">http://www.dundee.ac.uk/library/graphics/dundee-shield-
    16.ico</Image>

</OpenSearchDescription>
Browser search plugins

… continued

```html
<Url type="text/html"
    template="http://library.dundee.ac.uk/F/?func=find-e&amp;adjacent=N&amp;request={searchTerms}&amp;find_code=FIND_WRD&amp;dunsid=IE7search"/>

• URL for performing the search
• Encode all ampersands (change & to &amp; )
• `{searchTerms}` is a placeholder for the user’s search terms
• For Aleph and MetaLib, strip out the session ID after /V/ or /F/
• We can track usage using our fake form element, “dunsid”.
```

Image height="16" width="16" type="image/x-icon">http://www.dundee.ac.uk/library/graphics/dundee-shield-16.ico</Image>
Browser search plugins

- Save your XML service definition on a web server
- Offering the search service to users via a web page:

You then need to make the XML file available on a web server, and provide links to the public to allow users to install it. Of course you may be able to persuade your computer services department to install it on all university PCs. This example is from one of our web pages: http://www.dundee.ac.uk/library/search/search-tools.php
Browser search plugins

- Offering the search service to users via a web page:

<a href="#" onclick="window.external.AddSearchProvider('http://www.dundee.ac.uk/library/itsupport/ie7search.xml')">
Install Library catalogue search provider</a>

You need a little JavaScript for the links which install the search plugin.
Browser search plugins

- Autodiscovery

It’s also possible to make the browser aware of the search plugin when the user visits your catalogue. In this example the browser has detected that there is a search plugin for our library catalogue available.
Browser search plugins

- Autodiscovery:

**meta-tags:**

```html<br>
<link
    title="UoD Library catalogue"
    rel="search"
    href="http://www.dundee.ac.uk/library/itsupport/ie7search.xml"
    type="application/opensearchdescription+xml"
/>
```

All you have to do is add a special `<link>` tag to the `<head>` section of the page, for example by modifying your meta-tags file in the Aleph OPAC.
• Low take-up in 2007/8 academic year
• Greatly increased usage in 2008/9
• Does this reflect the habits of the new intake / user education?
Browser search plugins for MetaLib

- Just as easy to define manually as for Aleph
- Consult the Deep Linking Guide
- Scripts available on EL Commons so you can allow users to install plugins for their own quicksets

http://www.exlibrisgroup.org/display/MetaLib/Search+provider+browser+plugins+for+MetaLib+quicksets
Here is an example of a link from a subject heading in the catalogue.
The standard Aleph services menu will open, offering searching and browsing within Aleph, or searching via Google or other services.
We have added our MetaLib quicksets to the screen.
Here is the search being performed in MetaLib.
Pushing search terms to MetaLib

**tab_z121:**

<table>
<thead>
<tr>
<th>Quickset ID</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOOGLE</td>
<td><a href="http://www.google.com/search?q=$0100&amp;meta=lr%3D%26hl%3Den">http://www.google.com/search?q=$0100&amp;meta=lr%3D%26hl%3Den</a></td>
</tr>
<tr>
<td>QS000000271</td>
<td><a href="http://cross-search.dundee.ac.uk/V/?func=quick-1-check1&amp;mode=simple&amp;find_request_1=$0100&amp;group_number=000000271&amp;force_login=Y&amp;dunsid=aleph">http://cross-search.dundee.ac.uk/V/?func=quick-1-check1&amp;mode=simple&amp;find_request_1=$0100&amp;group_number=000000271&amp;force_login=Y&amp;dunsid=aleph</a></td>
</tr>
<tr>
<td>QS000000456</td>
<td><a href="http://cross-search.dundee.ac.uk/V/?func=quick-1-check1&amp;mode=simple&amp;find_request_1=$0100&amp;group_number=000000456&amp;force_login=Y&amp;dunsid=aleph">http://cross-search.dundee.ac.uk/V/?func=quick-1-check1&amp;mode=simple&amp;find_request_1=$0100&amp;group_number=000000456&amp;force_login=Y&amp;dunsid=aleph</a></td>
</tr>
</tbody>
</table>

Just add the quicksets to tab_z121 in Aleph, using the Deep Linking Guide to help you with the syntax.

We have given them rather cryptic names based on the quick set IDs in MetaLib, but you could give them meaningful names. We have used cryptic names because we have a script which automates the process, but that falls outside the scope of “easy OPAC enhancements”! If you don’t change your quicksets often you don’t really need a script.
Pushing search terms to MetaLib

**tab_z121:**

GOOGLE http://www.google.com/search?q=$0100&meta=lr%3D%26hl%3Den

QS000000271 http://cross-search.dundee.ac.uk/V/?func=quick-1-check1&mode=simple&find_request_1=$0100&group_number=000000271&force_login=Y&dunsid=aleph

QS000000456 http://cross-search.dundee.ac.uk/V/?func=quick-1-check1&mode=simple&find_request_1=$0100&group_number=000000456&force_login=Y&dunsid=aleph

**Placeholder for user’s search terms**
Pushing search terms to MetaLib

**service-engine:**

```html
<td class=td1 id=bold width=17%>
  <select name="engine">
    <option value=google>Google</option>
    <option value=QS000000271>MetaLib</option>
  </select>
</td>
```

Add quicksets to existing menu, or add a separate menu if you wish.

Then add them to service-engine.
External links simplified

This is what you get by default when following an external link in Aleph. Obviously in the UK we want to cut out the US copyright warning!
External links simplified

media-permission

• Can strip out most of the stuff from this file

• Still left with one more page than is really needed

• N.B. this page is possibly needed if you are using ADAM
External links simplified

<!-- filename: media-permission -->
<html>
<head>
<title>External link</title>
<script language="Javascript">
<!--
function goToLink() {
  // Get the placeholder into a string.
  // Fortunately it never contains single quotes.
  var s = '$0100';

  // Extract the part of the placeholder between
  // the first pair of double quotes. This happens
  // to be the destination URL.
  s = s.substring(s.indexOf('"')+1);
  s = s.substring(0,s.indexOf('"'));

  continued…

Here is how we have altered media-permission to remove the unnecessary intermediate page. And it works for users without JavaScript too.
External links simplified

… continued:
   // Move to this URL.
   window.location.href = s;

-->
</script>
</head>
<body onload="javascript:goToLink()">
<noscript>$0100 Please click the icon to access the external site.</noscript>
</body>
</html>
External links simplified

**media-display-exec:**

```html
<!-- filename: media-display-exec-->
<html>
<head>
<title>&f-base - External Link</title>
<include>meta-tags
<meta http-equiv="REFRESH" content="0; URL=$0100">
</include>
<body onload=window.location="$0100">
</body>
</html>
```

To ensure it works for users without JavaScript, you should add the highlighted section to `media-display-exec`. 
EndNote download fix

- Try downloading a batch of records in ISI ResearchSoft Tagged Output Format

**E-Mail or Save Selected Records**

You have 2 options:

- Send the records by e-mail (by entering an e-mail address)
- Save the records locally on your PC (by leaving the e-mail address blank)

The records will be formatted according to your choice.
Unfortunately the user will get a text file appearing in the browser!
EndNote download fix

- Add to `$httpd_root/conf/mime.types`:
  
  ```
  application/x-research-info-systems   end
  ```

- Alter `$httpd_root/conf/httpd.conf`:
  
  ```
  # Added "end" for EndNote files; MEP 8/06/2009
  <Files ~ "\.\.(gif|jpe?g|png|css|sav|end)$">
    Header unset Cache-Control
    Header unset Expires
    Header unset Pragma
  </Files>
  ```

It is easily fixed by altering the Apache configuration.
Once that is done you get the file downloading direct into EndNote, as it ought to.
<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Title</th>
<th>URL</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown</td>
<td>2001</td>
<td>Perl: the complete reference</td>
<td><a href="http://libproxy.dunde">http://libproxy.dunde</a>...</td>
<td><a href="http://libproxy.dunde">http://libproxy.dunde</a>...</td>
</tr>
<tr>
<td>Harorgan</td>
<td>2001</td>
<td>Custom CGI scripting with Perl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Holzmer</td>
<td>2001</td>
<td>Perl Black Book</td>
<td><a href="http://libproxy.dunde">http://libproxy.dunde</a>...</td>
<td></td>
</tr>
<tr>
<td>Lee</td>
<td>2001</td>
<td>Malcolm X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comerdy</td>
<td>2000</td>
<td>Object-oriented Perl: a comprehensive guide to...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sebastian</td>
<td>2000</td>
<td>A little book on Perl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well</td>
<td>2000</td>
<td>Programming Perl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perl</td>
<td>1999</td>
<td>Louise Mathesdotter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schiell</td>
<td>1998</td>
<td>Perl pronto! A jump-start guide to programming in...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ehlke</td>
<td>1998</td>
<td>Perl by example</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deep</td>
<td>1996</td>
<td>Developing CGI applications with Perl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Till</td>
<td>1996</td>
<td>Teach yourself PERL in twenty-one days</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vromans</td>
<td>1998</td>
<td>Perl 5 desktop reference</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions?

Ideas?
OPAC scripting enhancement

- IGeLU enhancement request no. 1764 – Server-side scripting support for OPAC
- Not for easy OPAC enhancements …
- … but makes the hard things possible
- Please vote!
Here is an example of a **difficult** OPAC enhancement! See it live for our periodical holdings at http://library.dundee.ac.uk/
**Record 18 out of 480**

- **Format**: Periodical, Electronic resource, Text
- **Holdings**: Location
- **Main Entry**: Great Britain, Office of Population Censuses and Surveys
- **Title**: Population Trends
- **Imprint**: Stationery Office
- **General note**: Main Library copies Not for loan
- **Subject**: Population statistics
  - Social Sciences General
  - School of Nursing and Midwifery
- **Continues**: Registrar General's Quarterly Return for England and Wales

**Holdings summary**: 
- **1993-2007**
  - **1994-**
  - **1995-2004**
  - **2003-2004**
- **1997-**

**Electronic link**

**Holdings chart**

**ISSN**

**Record number**

**Holdings**