Following the Yellow Brick Road to Simplified Link Management

Developing a database to maintain my library's links made everyone's lives easier and provided users with a valuable resource.

BY NICOLE C. ENGARD

"You want me to add *how many* links?"

"Why doesn't the description for Google on this page match the description we have on the other pages?"

These were questions that my Research Links Team frequently heard me asking. My role as Web manager at Jenkins Law Library in Philadelphia put me in charge of all additions and changes to the library Web site (http://www.jenkinslaw.org), which included daily edits to the Research Links pages. Jenkins is no different from other libraries; we too are providing more information via the Internet to keep up with the demands of our patrons. In my opinion, our Research Links are the biggest asset of our site. All legal (http://legallinks.jenkinslaw.org) and nonlegal (http://nonlegallinks.jenkinslaw.org) links are reviewed by librarians before they are approved for inclusion, to ensure that the information we are providing to our patrons is accurate, current, objective, and of good quality.

Jenkins is the oldest law library in America, and it has a reputation for offering great content not only to local attorneys, but also to the entire legal research community. We had a lot of great links—more than 250 pages with 1,500-plus links—but they were all coded in straight HTML, which meant that only I could make the necessary edits. It also meant that, with so much data, things were inconsistent from page to page. In June of 2004 I decided that I had had enough; it was time for a change.

»

Pay No Attention to the Woman Behind the Curtain

My main goal was to automate the process by which research links were added so pages could be generated dynamically and so I wouldn't have to always do daily updates. However, it was very important to me that I do this without changing the way the users interacted with the site. As in *The Wizard of Oz*, I didn't want anyone to know there was someone "behind the curtain" manipulating things.

I decided to start by reviewing some free, prepackaged link modules and quickly realized that there was nothing already written that would allow us to keep our Research Links organized the way they were. My long-term goal was to convert the entire Web site from HTML to the general-purpose scripting language PHP, so we would be able to offer more dynamic content to our users and to allow for more use of templates throughout the site. For this reason, I decided to use a combination of PHP and MySQL, an open source database management system, for this project. Having made this decision, I created a link module from scratch.

By storing the data in a MySQL database I had hoped to not only make our work easier, but also to make our data more reliable. I wanted to be able to have only one occurrence of each link in the database, even if that one link appeared on several different Web pages. An example would be our link to Google Groups, which was on our Mailing Lists & Newsgroups page as well as our Online Directories page. Prior to this project, we would have had to remember that this link was on multiple pages and that all of those pages needed to be edited when the URL or description changed. This also applied to pages that appeared under more than one category (i.e., Online Directories appears under both Business and General Reference).

The librarians on the Research Links Team were used to e-mailing me with their updates (the link's title, description, URL, the page the link was on, and possibly what heading the link should appear under). I wanted to design it so that a librarian would type this information into a form (instead of an e-mail), and the data would be entered into the database.

Mapping Out Our Own Yellow Brick Road

The hardest part of any database project is creating tables that will store the data in the most effective way. So I had to design a path for our data flow—the yellow brick road we'd follow, if you will. In our case we had a lot of overlapping data and wanted to stamp out redundancy as much as possible; this would eliminate a lot of extra maintenance in the process. In the beginning I had only two tables: one for links and one for categories (or pages). This layout did not give us the flexibility we needed, so I expanded the two tables into six:

1. The Categories table holds the information about each individual page.

2. The Subcategories table links Categories together to create a parent–child relationship (i.e., Online Directories is a child of both Business and General Reference).

3. The Headings table holds the bolded subheadings that appear on some of our pages.

4. The Links table holds the data related to the link (except the page it should appear on).

5. The States table holds the state name and abbreviation for all 51 U.S. states (including the District of Columbia) for our U.S. State Research Links pages.

6. The Linked table ties all of the above tables together by matching their primary keys.

As soon as I was sure that the table layout could accommodate our data, I designed a module through which the librarians could make updates and additions to the Research Links database. There are several different ways to handle table data when inserting information into a database. Up to this point I had only written single-page forms that *either* entered data into a database *or* e-mailed it to a specific person. With the Research Links I needed to create a multi-page form that would enter data into the database *and* e-mail confirmations to me and the person who made the change.

Knowing this, I decided that the best way to handle things was to write one page for adding new links, one for adding new pages, one to edit links, and one to edit pages. These four pages would make up the Research Links Indexing Module.

From the main page of the Indexing Module the librarians could choose to either add a link or page, or edit a link or page. Adding a link is a three-step/four-page process:

1. Enter all of the data for the link, including title, byline (if there is one), URL, and description. This is also where the librarian enters her e-mail address, which will be used for confirmation e-mails and broken link reports.

2. Next, the librarian chooses which pages (any or all) she would like to have her link on.

3. If warranted, the next page lists bolded subheadings that she has to choose from.

4. Finally, a page appears to confirm that the link has been added.

I decided that it was easiest for me to handle these multi-page forms by using the predefined $_GET array to pass the form values through the URL. This way data would be entered into the database every time the librarian hit a Submit

»

MySQL Tables Designed for Our Research Links

Categories

Field	Type
catid	int(11)
title	varchar(255)
pgtitle	varchar(255)
descript	varchar(255)
pgblurb	text
date	datetime
email	varchar(100)
approved	tinyint(4)
edited	datetime

Subcategories

Field	Type	Links to
key	int(11)	
parent	int(11)	categories -> catid
catid	int(11)	categories -> catid
headid	int(11)	headings -> headid

Headings

Field	Type	Links to
headid	int(11)	
catid	int(11)	categories -> catid
title	varchar(255)	
blurb	text	
display	tinyint(4)	

Links

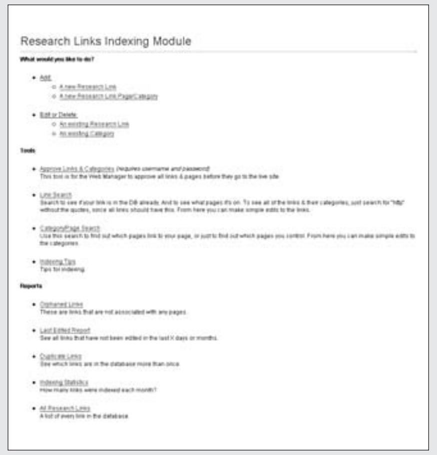
Field	Type
linkid	int(11)
title	varchar(255)
url	text
blurb	text
linkby	varchar(255)
approved	int(4)
date	datetime
email	varchar(100)
edited	datetime

States

Field	Type
stateid	int(11)
state	varchar(50)
abbrev	varchar(5)
edited	datetime

Linked

Field	Type	Links to
key	int(11)	
linkid	int(11)	links -> linkid
catid	int(11)	categories -> catid
headid	int(11)	headings -> headid
stateid	tinyint(4)	states -> stateid


Research Links Indexing Module

What would you like to do?

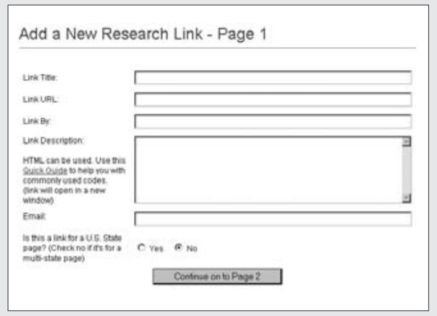
- Add:
  - A new Research Link
  - A new Research Link Page/Category
- Edit or Delete:
  - An existing Research Link
  - An existing Category

Tools

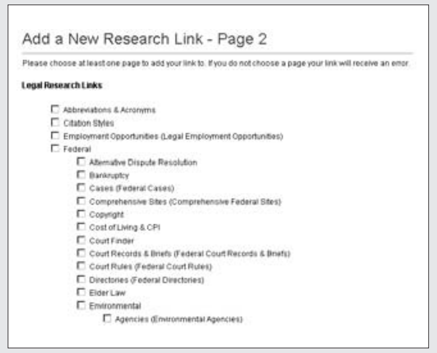
- Approve Links & Categories *(requires username and password)*
  This tool is for the Web Manager to approve all links & pages before they go to the live site.
- Link Search
  Search to see if your link is in the DB already. And to see what pages it's on. To see all of the links & their categories, just search for "http" without the quotes, since all links should have this. From here you can make simple edits to the links.
- Category/Page Search
  Use this search to find out which pages link to your page, or just to find out which pages you control. From here you can make simple edits to the categories.
- Indexing Tips
  Tips for indexing.

Reports

- Orphaned Links
  These are links that are not associated with any pages.
- Last Edited Report
  See all links that have not been edited in the last X days or months.
- Duplicate Links
  See which links are in the database more than once.
- Indexing Statistics
  How many links were indexed each month?
- All Research Links
  A list of every link in the database.


Add a New Research Link - Page 1

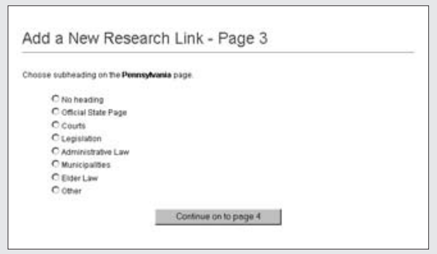
Link Title:	
Link URL:	
Link By:	
Link Description: HTML can be used. Use this Quick Guide to help you with commonly used codes. (link will open in a new window)	
Email:	
Is this a link for a U.S. State page? (Check no if it's for a multi-state page)	Yes [✓ No]


Add a New Research Link - Page 2

Please choose at least one page to add your link to. If you do not choose a page your link will receive an error.

Legal Research Links

- ☐ Abbreviations & Acronyms
- ☐ Citation Styles
- ☐ Employment Opportunities (Legal Employment Opportunities)
- ☐ Federal
  - ☐ Alternative Dispute Resolution
  - ☐ Bankruptcy
  - ☐ Cases (Federal Cases)
  - ☐ Comprehensive Sites (Comprehensive Federal Sites)
  - ☐ Copyright
  - ☐ Cost of Living & CPI
  - ☐ Court Finder
  - ☐ Court Records & Briefs (Federal Court Records & Briefs)
  - ☐ Court Rules (Federal Court Rules)
  - ☐ Directories (Federal Directories)
  - ☐ Elder Law
  - ☐ Environmental
    - ☐ Agencies (Environmental Agencies)


Add a New Research Link - Page 3

Choose subheading on the **Pennsylvania** page.

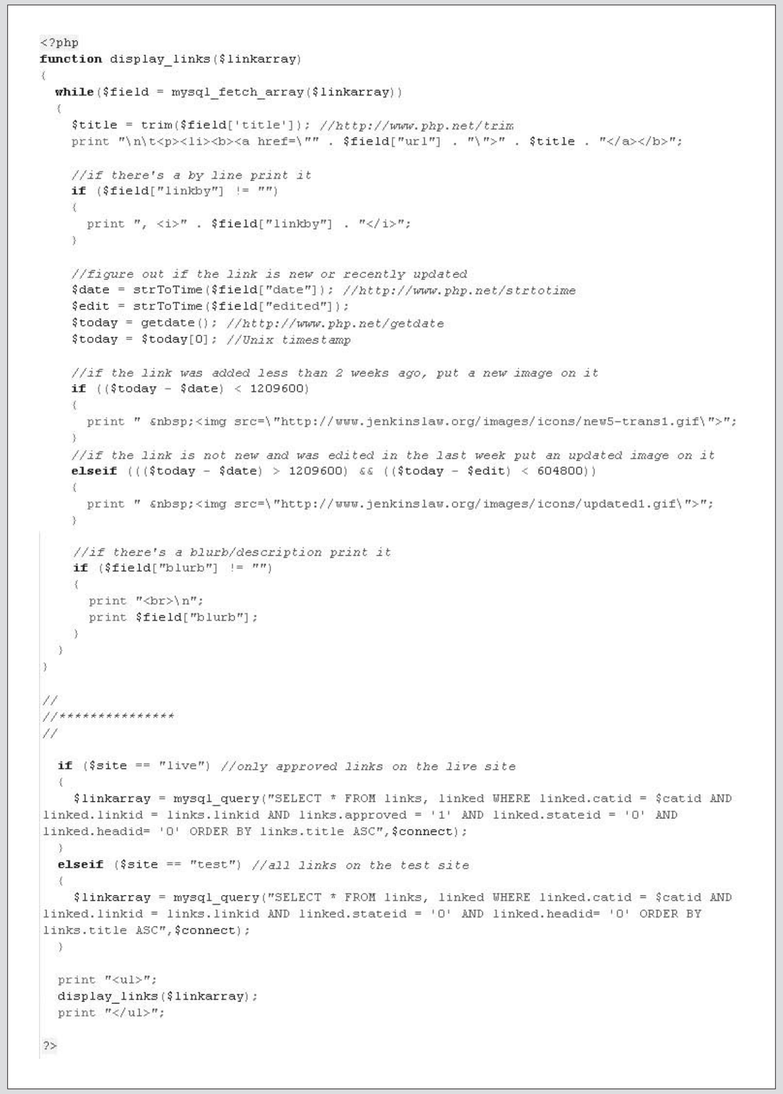
- ○ No heading
- ○ Official State Page
- ○ Courts
- ○ Legislation
- ○ Administrative Law
- ○ Municipalities
- ○ Elder Law
- ○ Other

Librarians start at the main page of the indexing module and choose whether to add or edit either links or pages. From there, other screens prompt them to make the necessary choices.

button and I could use that data on the next page.

Lions and Tigers and Other Obstacles, Oh My!

Now that I had an indexing module in place, I presented my idea to the librarians on the Research Links Team. As with any new development, the initial reaction was not overwhelmingly positive. They were afraid of running into trouble along the way. They wanted to know how this was going to affect their work flow and that of the library patrons. Some of our pages were used as the basis for our Continuing Legal Education (CLE) class booklets; did this mean that the books would have to be re-written? There was also the usual nervousness associated with doing something in a new way. Most importantly, the librarians wanted to make sure that there was still going to be some sort of quality control in place to ensure that the pages would still look the same.



```
<?php
function display_links($linkarray)
{
  while($field = mysql_fetch_array($linkarray))
  {
    $title = trim($field['title']); //http://www.php.net/trim
    print "\n\t<p><li><b><a href=\"" . $field["url"] . "\">" . $title . "</a></b>";

    //if there's a by line print it
    if ($field["linkby"] != "")
    {
      print ", <i>" . $field["linkby"] . "</i>";
    }

    //figure out if the link is new or recently updated
    $date = strToTime($field["date"]); //http://www.php.net/strtotime
    $edit = strToTime($field["edited"]);
    $today = getdate(); //http://www.php.net/getdate
    $today = $today[0]; //Unix timestamp

    //if the link was added less than 2 weeks ago, put a new image on it
    if (($today - $date) < 1209600)
    {
      print " &nbsp;<img src=\"http://www.jenkinslaw.org/images/icons/new5-trans1.gif\">";
    }
    //if the link is not new and was edited in the last week put an updated image on it
    elseif ((($today - $date) > 1209600) && (($today - $edit) < 604800))
    {
      print " &nbsp;<img src=\"http://www.jenkinslaw.org/images/icons/updated1.gif\">";
    }

    //if there's a blurb/description print it
    if ($field["blurb"] != "")
    {
      print "<br>\n";
      print $field["blurb"];
    }
  }
}

//
//***************
//

  if ($site == "live") //only approved links on the live site
  {
    $linkarray = mysql_query("SELECT * FROM links, linked WHERE linked.catid = $catid AND
linked.linkid = links.linkid AND links.approved = '1' AND linked.stateid = '0' AND
linked.headid= '0' ORDER BY links.title ASC",$connect);
  }
  elseif ($site == "test") //all links on the test site
  {
    $linkarray = mysql_query("SELECT * FROM links, linked WHERE linked.catid = $catid AND
linked.linkid = links.linkid AND linked.stateid = '0' AND linked.headid= '0' ORDER BY
links.title ASC",$connect);
  }

  print "<ul>";
  display_links($linkarray);
  print "</ul>";

?>
```

The code above was used to display the bulleted list of links for the Research Links pages.

To ease their fears I decided that, after the librarians had submitted their requests, I would check everything before allowing our users to see the new links or pages on the live site. I chose to keep two copies of the database, one in-house and one on the live server, and I added a field to the Categories and Links tables titled "approved" to keep track of whether I had approved a link for the live site.

I needed a way to come back and review the links that librarians had entered, so I developed an "Approval Module." When a librarian enters the link data, the "approved" field is set to zero. My new Approval Module pulled all of the links and categories out of the database that had the "approved" field set to zero. From this form I could then edit, delete, or approve the links and pages for the live site. Once I had OK'd the new additions, the "approved" field was set to one, and an e-mail confirmation would be sent to me and to the librarian to let us know that the new data was now on the live site.

Lastly, we needed a way to decipher which pages were on our local test site and which were live so that we could view all links and pages in-house before they were approved for the public. I defined a variable ($site) at the top of the main Research Links page (index.php) to indicate whether the page was for the public ($site = "live";) or for us in-house ($site = "test";). If the page was on our local server, all links were displayed; if the page was live, then only the links with the "approved" field set to one would display. I did this by using a simple IF statement wrapped around the MySQL query that was pulling the links from the database.

Once the Indexing Module was complete it was time to focus on the pages our users were going to see. These pages consisted of five main parts: path, page title, optional opening paragraph, optional bolded subheadings, and links. I started by opening up the HTML of one of our Research Links pages that had all five of these parts and replacing the main pieces with variables from the database. This way I was able to see if our data would fit into the current layout. With little tweaking, it did. In the end, I was able to turn more than 250 pages into two templates: one for the U.S. States pages (because their formatting was so different from the other pages) and one for all other Research Links pages.

Gifts from the Wizard: Brain, Heart, and Courage

As soon as everything was in place and tested, it was time for the librarians to do their part. They had the task of taking more than 1,500 links and entering them all into the new database. We thought this would be a simple, albeit tedious, task, but it turned out that there were many factors that we hadn't considered.

We had been running weekly link checks on the site to make sure that all of our links were up-to-date. If there were problems, I would let the librarians know and they in turn would tell me how to fix them. What we hadn't realized was that we had only been checking for broken links—pages that no longer exist—but that didn't encompass all of the possible errors. Pages that had redirects or messages on them saying the site had moved were not showing up in our reports. In addition to the new URLs, the librarians also had to make sure that their descriptions still applied. Lastly, the librarians had to check to make sure that they were entering the link into the database only once. To ensure that we had only one description to make upkeep easier down the road, they had to search the whole Jenkins Law Library site to see where else the link appeared. In short, it was a very time-consuming process for all six librarians on the Research Links Team, and lasted about 3 months. But my librarians did it, as I knew they could. They had the brains, the heart, and the courage to make it work—all I had to do was show faith in them and point the way.

We came to realize that we needed a better link-checking method in place; something that would not only check the URLs, but also require us to come back and read over our links and pages to make sure that they were all in order. I wrote a report that would send out an e-mail once every quarter to everyone who had links in the database. The librarians were then expected to go through their individual reports and check every link for any necessary updates.

We decided to generate two other reports to help us maintain the data. One was a duplicate report that listed all links that were in the database more than once and showed which pages they were on. It ran from the following query:

```
SELECT a.linkid, a.url, a.title
FROM links a INNER JOIN links
b ON a.url = b.url GROUP BY
a.linkid, a.url, a.title HAVING
COUNT( b.linkid ) >1 ORDER
BY a.url, a.linkid ASC;
```

The other report shows "orphaned" links that were not assigned to any specific page, but were in the database, using the following query:

```
SELECT links.title, links.url,
links.linkby, links.linkid, links
.blurb FROM links LEFT JOIN
linked ON linked.linkid = links
.linkid WHERE linked.linkid IS
NULL ORDER BY links.title ASC;
```

I created this report so that someone could remove non-functioning links from a page but still keep them in the database so that they could make them live again when the site had been repaired.

There's No Place Like Home

On Oct. 4, 2004, almost 6 months after I first decided to automate our processes, our new Research Links database went live without a hitch. Since that time our index has grown from more than 1,500 links on about 250 pages to just over 2,000 links on about 300 pages. Despite the initial trepidation on the part of the librarians, I am constantly hearing from them how much they appreciate this development; adding and editing the Research Links pages have never been easier.

When I set out to develop this database my main goal was to make my life easier; the result, however, was that I made everyone's lives easier and was able to provide our users with a much more valuable resource in the process. While it felt as if we'd been on a long journey, all our adventures had been in our own building with our own staff. We'd never really left home, but had still learned a valuable lesson about our own abilities to create database-driven pages for the Jenkins Law Library Web site.

Nicole C. Engard is Web manager at Jenkins Law Library in Philadelphia. She holds a B.A in literary studies/computer programming from Juniata College in Huntingdon, Pa. She directs the development and maintenance for the Jenkins Web site, palawlibrary.com, and the staff intranets. She also guides and supports all other Web-related and database projects including PA Legislative Histories, Jenkins Electronic Resources Network, JAC Plus (Jenkins Automated Catalog), and the Jenkins Court Records & Briefs online indexes. Her e-mail address is nengard@jenkinslaw.org.

Further Reading

Sklar, David and Adam Trachtenberg. *PHP Cookbook*. Sebastopol, Calif: O'Reilly, 2002.
Williams, Hugh E. and David Lane. *Web Database Applications with PHP and MySQL, 2nd Edition*. Sebastopol, Calif: O'Reilly, 2004.
Dev Articles–PHP: http://www.devarticles.com/c/b/PHP
Dev Shed Web Development Forums: http://forums.devshed.com
Jenkins Web Development Research Links: http://www.jenkinslaw.org/researchlinks/index.php?rl=226
MySQL Manual: http://dev.mysql.com/doc/mysql/en
Official PHP Site: http://www.php.net
PHP Manual: http://www.php.net/manual/en