# dbWiz: Open Source Federated Searching for Academic Libraries

Calvin Mah, *Simon Fraser University Library, Burnaby, Canada*
Kevin Stranack, *Simon Fraser University Library, Burnaby, Canada*

The current issue and full text archive of this journal is available at:
http://www.emeraldinsight.com/0737-8831.htm

## Introduction

Faced with the choice between multiple subscription databases, each with a different interface and search functions, and the simplicity of Google, college and university students are increasingly finding their research materials on the open web. Federated searching provides one way that academic libraries can begin to win back some of these novice users, and ensure they are finding the highest quality information available. dbWiz is an open source federated searching tool currently being developed at the Simon Fraser University Library, funded by nine partner post-secondary institutions. The SFU Library has been developing open source software for a several years, including the reSearcher software suite, which features a link resolver (GODOT), a serials management knowledge base (CUFTS), electronic resource management tools, and more. This article provides an overview of the dbWiz development process, the functionality of the software, and discusses some of the benefits and challenges faced by the project.

## Federated Searching

Federated searching, also known as metasearching, broadcast searching, cross searching, and a variety of other names, is the ability to search multiple information resources from a single interface and return an integrated set of results. Although aspects of this kind of shared searching has existed for some time (especially with Z39.50 catalogue searching), the explosion of online content and the rise of Google as the dominant web-based search tool has made the development of this kind of searching more important than ever.

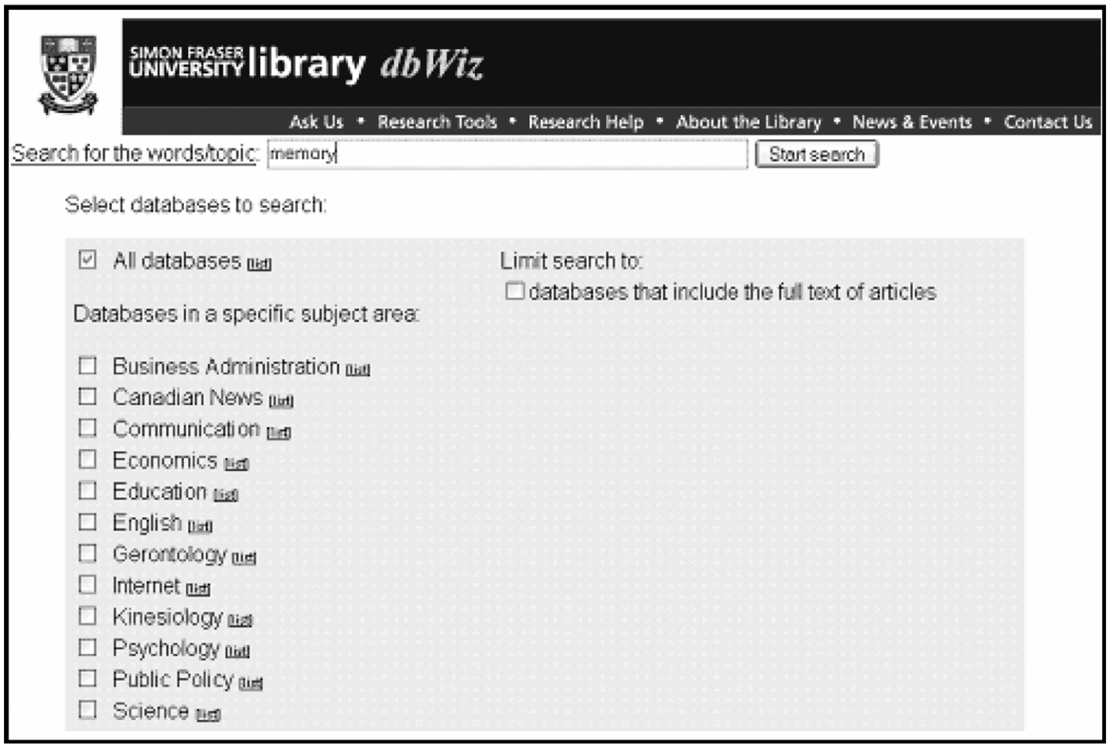
Google has set a new standard for fast and easy to use searching that brings back “good enough” results almost every time. Increasingly, many library users are relying solely on this powerful search engine to the neglect of the valuable subscription collections libraries provide (Luther, 2003). By paying attention to what Google is doing right, by “breaking down the information silos” (Webster, 2004a), and by examining the technology that is now available to make systems interoperate more efficiently than ever before, libraries can make sure that “good enough” results get even better through the use of federated search tools such as dbWiz.

## Project Background

dbWiz was originally developed to help students determine the best starting point for their research. Within the Council of Prairie and Pacific University Library (COPPUL) consortium, there was concern that novice library users would be overwhelmed by the growing number of online resources available, and required an online tool to assist them in the absence of a librarian.

Building on the Simon Fraser University Library's experience with Z39.50 searching gained in the development of the GODOT link resolver and interlibrary search and request system, we were able to develop dbWiz version one.

By entering a search term into dbWiz version one (see Figure 1), users would be presented with a list of databases with the number of results that would be found by searching the native interface.


SIMON FRASER UNIVERSITY library dbWiz
Ask Us • Research Tools • Research Help • About the Library • News & Events • Contact Us

Search for the words/topic: memory [Start search]

Select databases to search:

☑ All databases [list]

Limit search to:

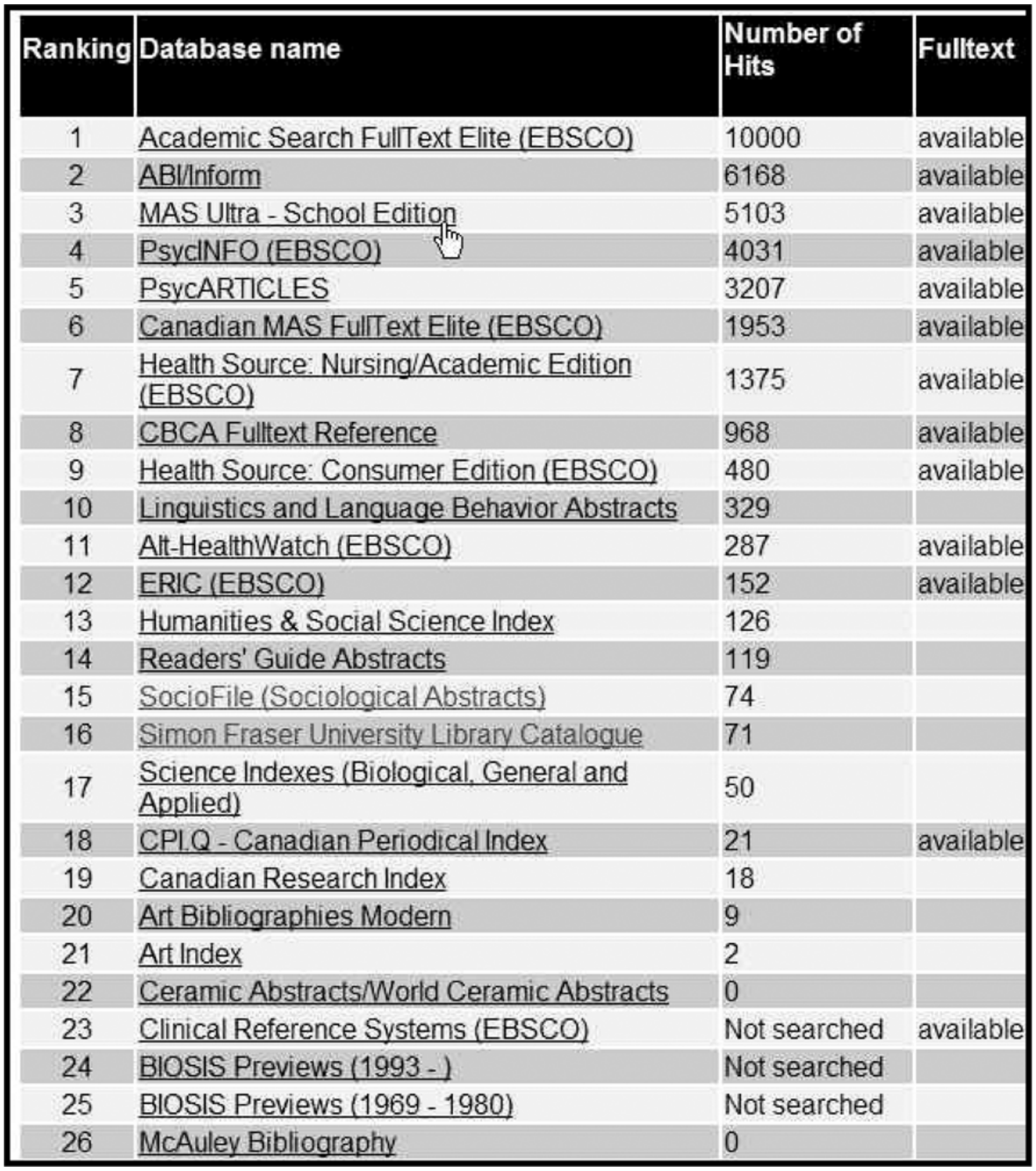
☐ databases that include the full text of articles

Databases in a specific subject area:

☐ Business Administration [list]
☐ Canadian News [list]
☐ Communication [list]
☐ Economics [list]
☐ Education [list]
☐ English [list]
☐ Gerontology [list]
☐ Internet [list]
☐ Kinesiology [list]
☐ Psychology [list]
☐ Public Policy [list]
☐ Science [list]

Figure 1: dbWiz version one search interface

For example, a dbWiz search for "memory" would return a ranked list of resources with a hit count and the availability of full-text (see Figure 2).



| Ranking | Database name | Number of Hits | Fulltext |
|---|---|---|---|
| 1 | Academic Search FullText Elite (EBSCO) | 10000 | available |
| 2 | ABI/Inform | 6168 | available |
| 3 | MAS Ultra - School Edition | 5103 | available |
| 4 | PsycINFO (EBSCO) | 4031 | available |
| 5 | PsycARTICLES | 3207 | available |
| 6 | Canadian MAS FullText Elite (EBSCO) | 1953 | available |
| 7 | Health Source: Nursing/Academic Edition (EBSCO) | 1375 | available |
| 8 | CBCA Fulltext Reference | 968 | available |
| 9 | Health Source: Consumer Edition (EBSCO) | 480 | available |
| 10 | Linguistics and Language Behavior Abstracts | 329 | |
| 11 | Alt-HealthWatch (EBSCO) | 287 | available |
| 12 | ERIC (EBSCO) | 152 | available |
| 13 | Humanities & Social Science Index | 126 | |
| 14 | Readers' Guide Abstracts | 119 | |
| 15 | SocioFile (Sociological Abstracts) | 74 | |
| 16 | Simon Fraser University Library Catalogue | 71 | |
| 17 | Science Indexes (Biological, General and Applied) | 50 | |
| 18 | CPI.Q - Canadian Periodical Index | 21 | available |
| 19 | Canadian Research Index | 18 | |
| 20 | Art Bibliographies Modern | 9 | |
| 21 | Art Index | 2 | |
| 22 | Ceramic Abstracts/World Ceramic Abstracts | 0 | |
| 23 | Clinical Reference Systems (EBSCO) | Not searched | available |
| 24 | BIOSIS Previews (1993 - ) | Not searched | |
| 25 | BIOSIS Previews (1969 - 1980) | Not searched | |
| 26 | McAuley Bibliography | 0 | |

Figure 2: dbWiz version one search results

The link would then take the user to the search page for that resource, where they would re-enter their search term to view the results. Although we had successfully met the challenge posed to us the previous year, we were somewhat disappointed by the response from the wider community. While we may have thought that this would a valuable addition to the online research toolkit, our users did not. We quickly realized that what people really wanted were the search results, from one search interface, delivered in a unified result set – in short, full federated searching.

## Moving to Version Two

Based on a review of several available federated searching tools (including dbWiz version one and other commercial products), we developed a seven-point list of key features that any successful federated search tool we could create would need to have. These included:

1. Retrieve and display integrated search results.
2. Splitting the results coming from Proquest by database (ABI/Inform, CBCA, and others). dbWiz One was unable to do this.
3. Provide direct linking to the full-text content or to the citation.
4. Sort results.
5. Allow for searching by author title and subject (when available for that resource).
6. Ability to limit the results to academic materials, full-text, and/or by date.
7. Add Boolean searching.

While this was an ambitious step from the original dbWiz project, we were confident that with funding, we would be able to meet the challenge. Based on the estimated time to implement each feature, we developed a budget for moving the project forward. The next step was to raise the money.

## Proposal

With dbWiz version one successfully completed as a prototype, together with a logical list of functionality enhancements and a clear cost analysis, we approached other post-secondary libraries in western Canada for support in developing this new product. The result was the creation of the dbWiz partnership in the summer of 2004, with nine member institutions each contributing funds determined by the size of their organization. Each partner library is consulted throughout the development process, will have their own dbWiz profile set up for their library, and will receive one year of support. As part of the partnership, each member library will also pay a modest annual fee to ensure the ongoing maintenance of the project.

## Development Team

With funding in place, we were able to create a dbWiz development team to see the project through to completion. The team consists of five members, including programmers working full-time on the software and librarians taking on project responsibilities in addition to their other duties. The members include: a systems coordinator, responsible for overall project management; a lead programmer, responsible for the overall programming of the software, but focusing on the search processes and interface design; a second programmer working primarily on the development of the many search plugins; a systems librarian, providing research and analysis; and a support librarian, working with the partner libraries, gathering feedback, providing updates, and developing documentation and implementation assistance. The total time to move from the research stage to software implementation will be about 18 months. The diverse backgrounds and expertise of the different team members has kept the dbWiz project on track for development target dates and budget requirements, while rapidly building a highly functional product.

## Resource List

The initial list of priority resources for dbWiz version two consisted of the databases shared by all nine members. These included the major aggregators, e-journal resources, important periodical indexes, and e-book collections. As all of the partners were members of the COPPUL consortium, there were many resources that overlapped.

Once these were identified, smaller databases shared by the majority of partners were included, and finally important unique or local databases were added, such as library catalogues, local institutional repositories, Simon Fraser University Library's Editorial Cartoons database, the Vancouver Public Library's Historical Photograph collection, and others.

## Search Plugins

Once we had decided upon a target set of resources, we were able to begin programming for version two, focusing first on building the many search plugins. Written as small Perl modules, these plugins are necessary for dbWiz to communicate with the diverse range of online databases and to understand the results that are returned from the resources. Although the current plugins are written in Perl, it would also be possible to create new plugins in any language, such as java or python.

There are two main access mechanisms that the vendors support which allow dbWiz to interact with their databases. First, is by web access only. This describes the majority of the commercial resources we subscribe to. The vendor provides no access to their database other than through a web browser. In order for dbWiz to search this type of database, the search plugin simulates a person searching the database, but instead of results going to a web browser, the results are parsed by dbWiz.

Although this method works, it requires maintenance. If the vendor changes what the way the search results are displayed, the dbWiz plugin will require modification. The second access mechanism is through an application program interface, commonly known as an API. An API is a set of instructions on how to write your own software to connect to the vendor's database. These APIs are usually, but not limited to, Z39.50. This access method is the most reliable since it uses a documented interface from which to search and retrieve results. dbWiz crafts the search according to the rules of the API and retrieves the results likewise. By 2005, over 100 APIs had been written for dbWiz version two.

## Parasearch

dbWiz uses a parallel searching engine, called Parasearch. In order for the dbWiz user interface to run efficiently, the work of doing the actual searches using the plugins must be off loaded to a separate process. dbWiz communicates to the Parasearch server via the simple object access protocol (SOAP), a standard method of exchanging XML information over computer networks. SOAP is used because it is lightweight, based on XML, and uses HTTP, the same protocol as the worldwide web as it's transport.

A typical search goes through the following steps:

1. A typical user types in a search in the dbWiz user interface and clicks on "search";
2. dbWiz sends the search that the user entered, along with the resources that the user wishes to search, to the Parasearch server via a SOAP call;
3. The Parasearch server looks at each resource that it is asked to search and calls the appropriate search plugin;
4. Parasearch returns the search results to the dbWiz web server and logs out of the resource when there is only a limited number of concurrent users permitted;
5. dbWiz collects the search results and presents it to the user.

## Authentication

A typical installation of dbWiz searches licensed databases and resources. These resources are limited to the IP address of the subscribing institution. dbWiz uses EZproxy, a URL rewriting proxy server to search resources on another institution's behalf.

dbWiz uses IP address authentication to determine which partner institution a dbWiz user belongs to. If a dbWiz user's IP address does not match an address that belongs to a partner institution, the user will only be allowed to search databases and resources that are not licensed or restricted. A user searching dbWiz from a home internet connection, which will yield an

IP address which dbWiz does not recognize, will have to connect to his own partner institution's proxy server to search dbWiz.

## Search Process

As a federated search tool, dbWiz needs to translate the search syntax from a single interface into the search syntax of multiple interfaces. The search syntaxes of varying resources differ greatly. In order to support a minimum set of search functions available in dbWiz, we had to decide what features were common to enough of the resources we wanted to search and include them in dbWiz. As a result, dbWiz searches the following indexes: keyword, author, title and subject. In cases where a search target does not support searching via author, title or subject, dbWiz uses the keyword index as a default.

System response time is also a crucial consideration. For dbWiz to be a useful search tool, it must return results in a reasonable amount of time. dbWiz searches all the resources simultaneously and uses a timeout strategy for determining which results to include with the final set. Resources that take too long to respond are dropped from the result list. Needing to choose between completeness and responsiveness, we decided that the priority for our target users was speed.

dbWiz is built on the same shared host model as the other SFU Library open source products. In our initial implementation, dbWiz is hosted at SFU for nine partner institutions. dbWiz uses an IP address database to determine which site an incoming connection belongs to. A configuration database stores the customized settings and templates that each site has set. The user is presented with the profiles and resources that are only available to his current site.

## Search Interface

By February 2005, a new user interface had been created. Also written in Perl, the search interface runs in an Apache/Mod_Perl environment. A key consideration in the development of dbWiz is customization. Since dbWiz was initially hosted at Simon Fraser University Library for all of the partners, the interface design was an immediate priority. To address the issue of customization, the Perl Template Toolkit was selected as a user interface creation tool. Using the Template Toolkit allows each individual institution to customize the elements of the dbWiz user interface to their own liking.

In the current state of development, dbWiz can be searched in three different ways. First, users may select from a predefined category of resources (see Figure 3).

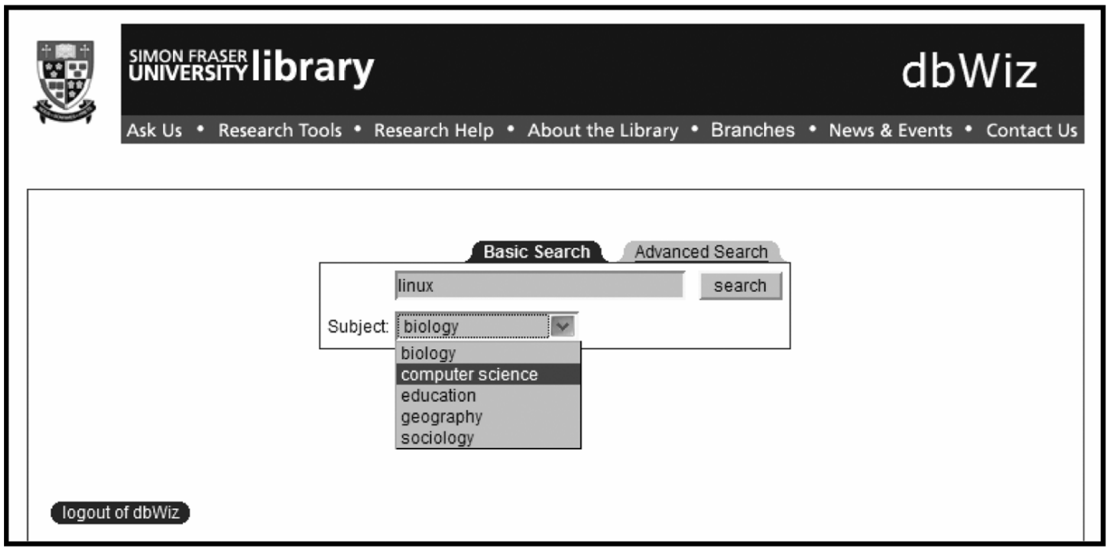
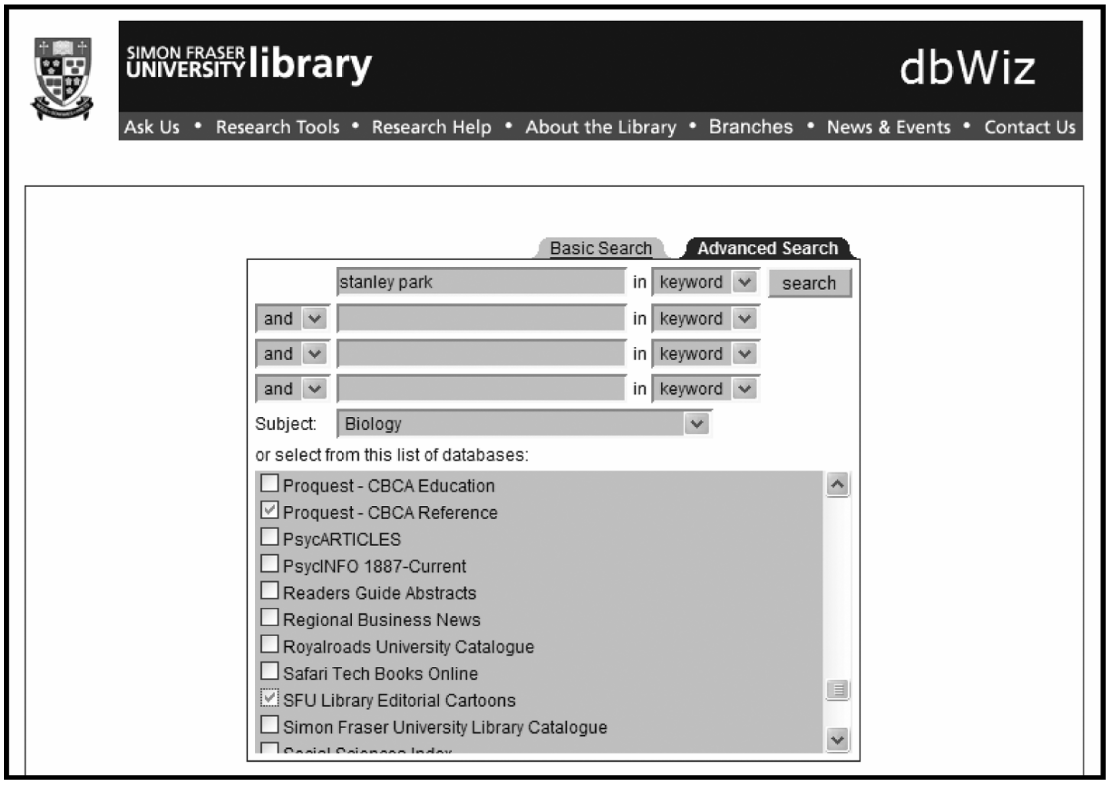

SIMON FRASER UNIVERSITY library
dbWiz
Ask Us • Research Tools • Research Help • About the Library • Branches • News & Events • Contact Us
Basic Search
Advanced Search
linux
search
Subject: biology
biology
computer science
education
geography
sociology
logout of dbWiz

Figure 3: Search categories

Choosing the category "Computer Science", for example, would allow for the simultaneous search of a variety of resources on the subject, including journal article collections, e-books, online encyclopaedias and dictionaries. By focusing on subject-specific collections whenever possible, libraries assist their users in increasing the relevance of their searching.

Second, users are able to select individual resources to search (see Figure 4). This feature will be of interest to a moderately experienced searcher who may have become familiar with particular resources from their previous searching experience. The Advanced interface consists of guided search fields and a list of searchable resources. While there is no theoretical limit on the number of resources that can be searched at once, response time will degrade with as selections are added.

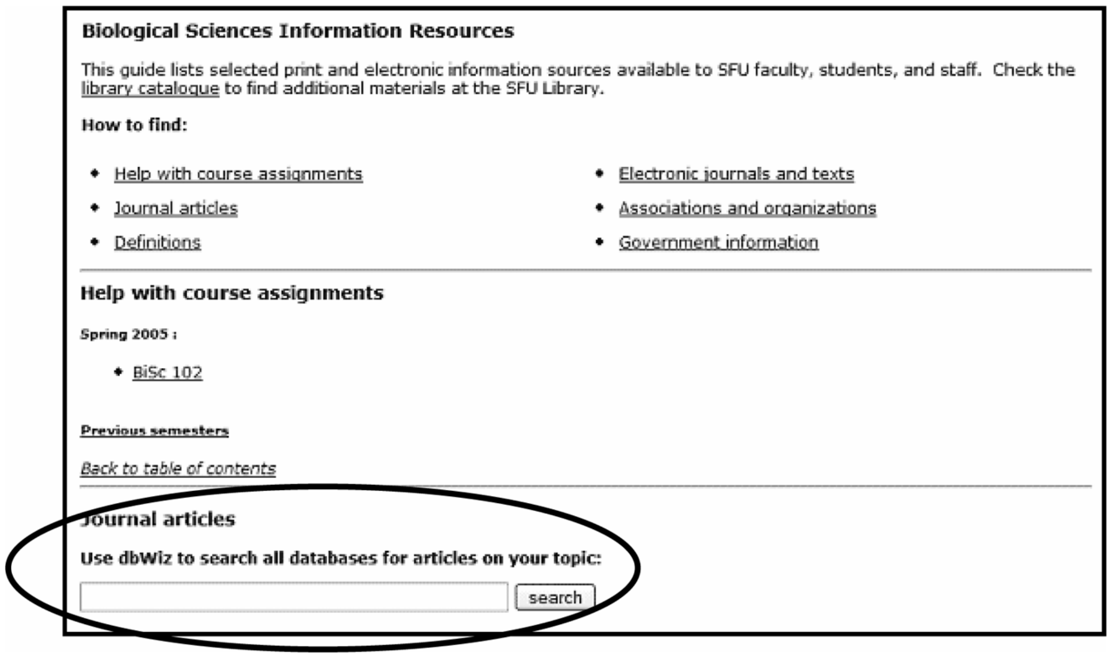

SIMON FRASER UNIVERSITY library
dbWiz
Ask Us • Research Tools • Research Help • About the Library • Branches • News & Events • Contact Us
Basic Search
Advanced Search
search

stanley park in keyword
and in keyword
and in keyword
and in keyword
Subject: Biology
or select from this list of databases:

- [ ] Proquest - CBCA Education
- [x] Proquest - CBCA Reference
- [ ] PsycARTICLES
- [ ] PsycINFO 1887-Current
- [ ] Readers Guide Abstracts
- [ ] Regional Business News
- [ ] Royalroads University Catalogue
- [ ] Safari Tech Books Online
- [x] SFU Library Editorial Cartoons
- [ ] Simon Fraser University Library Catalogue

Figure 4: Resource selection

Finally, it is also possible to embed a dbWiz search box directly into any web page. By placing dbWiz directly on a subject guide page, for example, users can quickly search the best resources for that discipline (see Figure 5).


**Biological Sciences Information Resources**

This guide lists selected print and electronic information sources available to SFU faculty, students, and staff. Check the library catalogue to find additional materials at the SFU Library.

**How to find:**

- Help with course assignments
- Journal articles
- Definitions
- Electronic journals and texts
- Associations and organizations
- Government information

**Help with course assignments**

**Spring 2005 :**

- BiSc 102

**Previous semesters**

*Back to table of contents*

**Journal articles**

**Use dbWiz to search all databases for articles on your topic:**

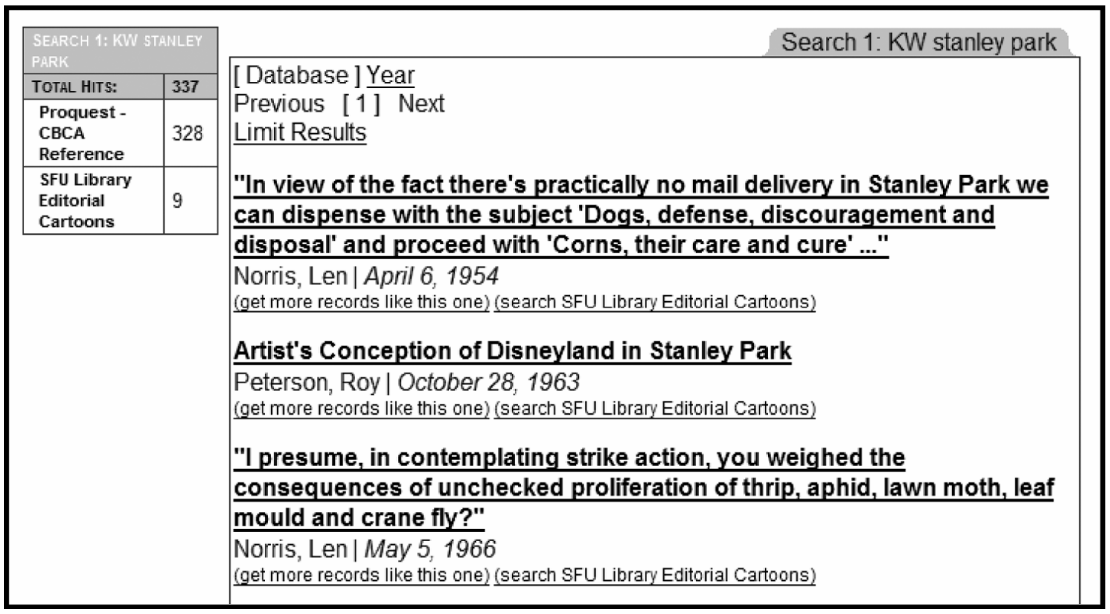
Figure 5: Embedded searching

This method can also allow libraries to create targeted searches, taking the search tool directly to the user (Abram, 2005), by placing a dbWiz link onto web pages outside of the library web site, such as university department pages or to individual courseware pages, with categories created for a specific discipline, course, or even assignment.

In all three search methods, users are able to use Boolean searching (AND, OR, NOT) whenever these are supported by the native interface. For any federated search tool, the functionality is always limited to what is available from the resources being searched. While this has led to some concern about “dumbed down” searches (Luther, 2003), federated searching provides a service that is central to what libraries have always done – bringing together resources into one place and making them easy to find (Webster, 2004b).

## Results Interface

Once the search has been initiated, a set of integrated results is returned (see Figure 6).



| SEARCH 1: KW STANLEY PARK | |
|---|---|
| TOTAL HITS: | 337 |
| Proquest - CBCA Reference | 328 |
| SFU Library Editorial Cartoons | 9 |

Search 1: KW stanley park

[ Database ] Year
Previous [ 1 ] Next
Limit Results

**"In view of the fact there's practically no mail delivery in Stanley Park we can dispense with the subject 'Dogs, defense, discouragement and disposal' and proceed with 'Corns, their care and cure' ..."**
Norris, Len | *April 6, 1954*
(get more records like this one) (search SFU Library Editorial Cartoons)

**Artist's Conception of Disneyland in Stanley Park**
Peterson, Roy | *October 28, 1963*
(get more records like this one) (search SFU Library Editorial Cartoons)

**"I presume, in contemplating strike action, you weighed the consequences of unchecked proliferation of thrip, aphid, lawn moth, leaf mould and crane fly?"**
Norris, Len | *May 5, 1966*
(get more records like this one) (search SFU Library Editorial Cartoons)

Figure 6: Result list

A library can determine how many records (10, 20, or more) are brought back from each resource. The greater number of records, however, the longer the search may take, and the longer the list of results to work through.

The results are displayed by whichever sorting option a library prefers (often date), but can be reordered by title, source, resource, and date. Determining relevance can be a significant challenge, as what is relevant to one user may not be relevant for another (Tennant, 2003). Currently, we are working on a relevance algorithm determined by a count of keywords found in the records, as our target users have become used to results sorted in this manner. Google does not return its results by date, source, or any other format other than relevance, and we need to ensure we are developing a system that will work the way our users expect.

Each record from a set of search results will display (when available) a brief amount of information, including title, author, date, journal title, and resource. Search limits currently include by date range, by academic, non-academic, or both, and by resources with full-text, without full-text, or both.

dbWiz also provides a direct link to the original record in the native interface or a link to an OpenURL resolver when a direct link is not provided. These links are crucial for users looking for fast and easy access to their research material. Ideally, each dbWiz record would contain a link directly back to the native record, however for resources that only allow only a limited number of concurrent users, we have opted to have dbWiz disconnect from the resource upon retrieving the results. This frees the resource for other users. In these cases, the OpenURL link will provide the most direct route to the original record.

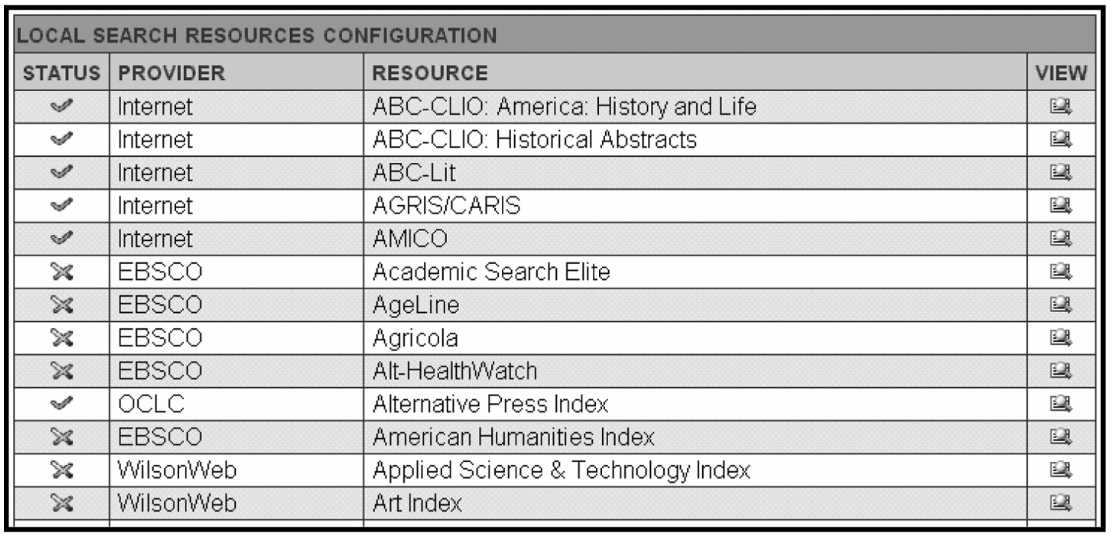
Search results also provide a link to retrieve the next set records for that resource ("get more records like this one") as well as a link to the native search interface for that resource (e.g. "search SFU Library Editorial Cartoons"), allowing for direct and more in-depth searching.

dbWiz also maintains a search history, allowing users to quickly re-create earlier searches. The search history also provides an instructional function, displaying the names and number of results from each resource searched. Although the results may be sorted in dbWiz in a variety of formats, the search history presents the original resources in a prominent and consistent manner. Because the interface has been created with templates and style sheets, each library has the ability to redesign their own results pages, allowing for a list that more closely resembles that of a simplified search engine or a detailed subscription database.

## Administrative Interface

The administrative interface allows libraries to create their own search categories and add or delete resources quickly and easily from their dbWiz profile. The administrative interface borrows the source code from GODOT, Simon Fraser University Library's open source link resolver. The parameters and settings of the administrative interface are stored in a postgreSQL database.

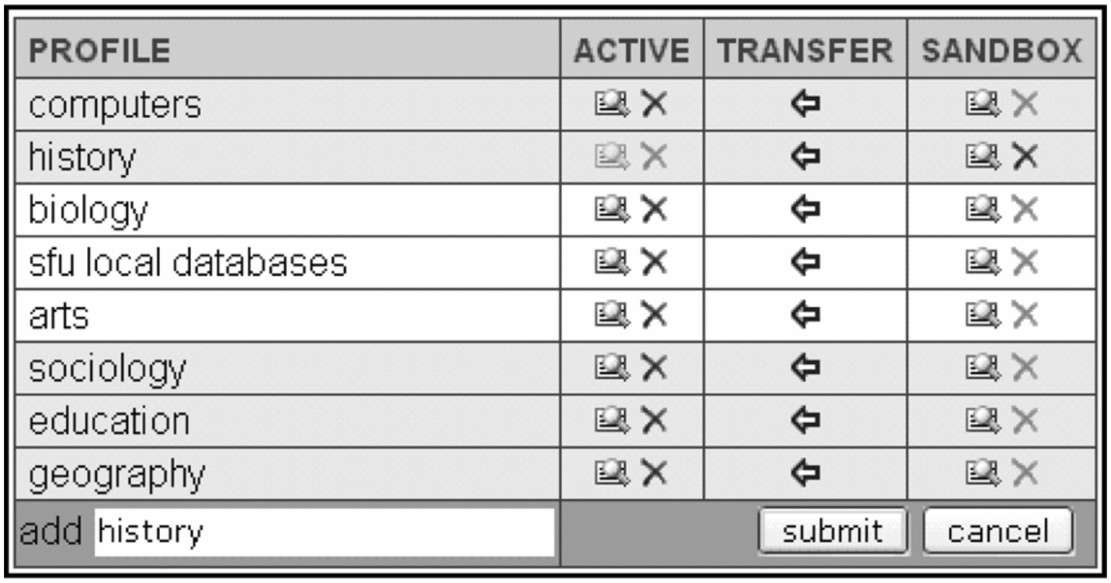
dbWiz not only needs to be fast and easy to use for students, but also for the library staff who will be adding new resources, creating and maintaining the search categories, and customizing the dbWiz interface. After logging into the administrative interface, library staff can access the global list of dbWiz resources, and activate the ones to include in their local collection (see Figure 7).



| LOCAL SEARCH RESOURCES CONFIGURATION | | | |
|---|---|---|---|
| STATUS | PROVIDER | RESOURCE | VIEW |
| ✓ | Internet | ABC-CLIO: America: History and Life | 🔍 |
| ✓ | Internet | ABC-CLIO: Historical Abstracts | 🔍 |
| ✓ | Internet | ABC-Lit | 🔍 |
| ✓ | Internet | AGRIS/CARIS | 🔍 |
| ✓ | Internet | AMICO | 🔍 |
| ✗ | EBSCO | Academic Search Elite | 🔍 |
| ✗ | EBSCO | AgeLine | 🔍 |
| ✗ | EBSCO | Agricola | 🔍 |
| ✗ | EBSCO | Alt-HealthWatch | 🔍 |
| ✓ | OCLC | Alternative Press Index | 🔍 |
| ✗ | EBSCO | American Humanities Index | 🔍 |
| ✗ | WilsonWeb | Applied Science & Technology Index | 🔍 |
| ✗ | WilsonWeb | Art Index | 🔍 |

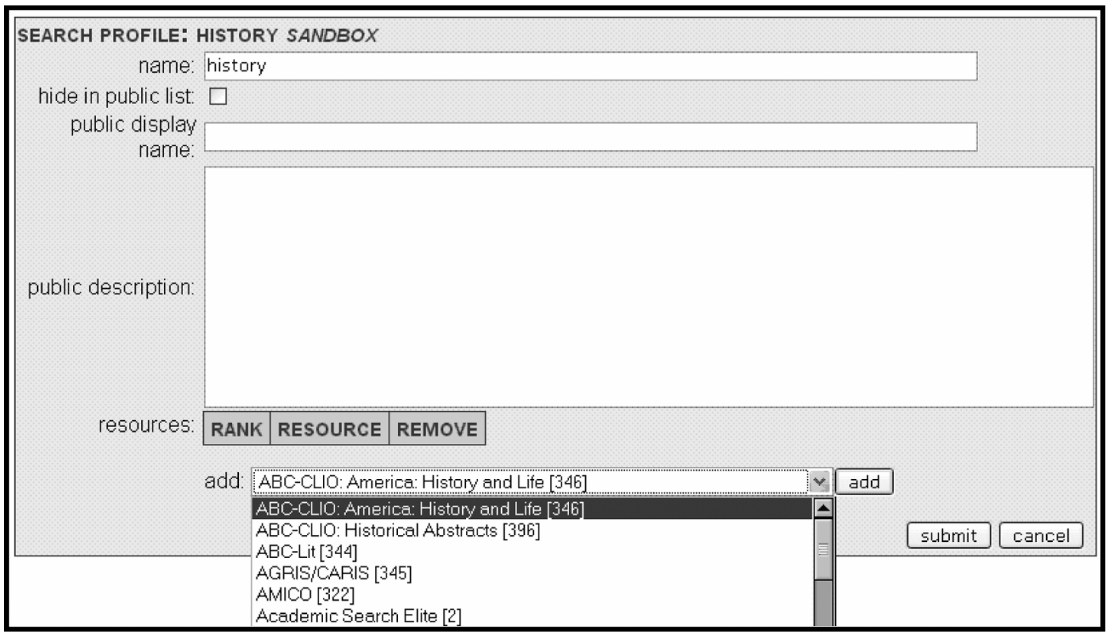
Figure 7: Activating local resources

Creating categories, whether based on a subject, a course, or even an assignment, can be done by simply entering in a new category term (see Figure 8) and adding the related resources from the local resource listing (see Figure 9).



| PROFILE | ACTIVE | TRANSFER | SANDBOX |
|---|---|---|---|
| computers | 🔍 ✗ | ⇦ | 🔍 ✗ |
| history | 🔍 ✗ | ⇦ | 🔍 ✗ |
| biology | 🔍 ✗ | ⇦ | 🔍 ✗ |
| sfu local databases | 🔍 ✗ | ⇦ | 🔍 ✗ |
| arts | 🔍 ✗ | ⇦ | 🔍 ✗ |
| sociology | 🔍 ✗ | ⇦ | 🔍 ✗ |
| education | 🔍 ✗ | ⇦ | 🔍 ✗ |
| geography | 🔍 ✗ | ⇦ | 🔍 ✗ |
| add history | | submit | cancel |

Figure 8: Creating search categories



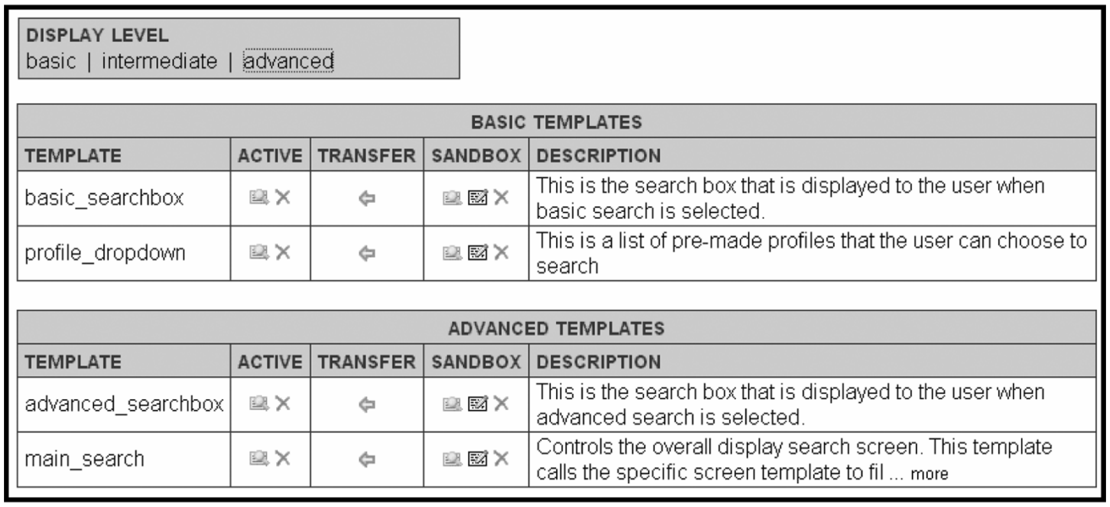
SEARCH PROFILE: HISTORY *SANDBOX*

| | |
|---|---|
| name: | history |
| hide in public list: | ☐ |
| public display name: | |
| public description: | |
| resources: | RANK \| RESOURCE \| REMOVE |
| add: | ABC-CLIO: America: History and Life [346] |

ABC-CLIO: America: History and Life [346]
ABC-CLIO: Historical Abstracts [396]
ABC-Lit [344]
AGRIS/CARIS [345]
AMICO [322]
Academic Search Elite [2]

Figure 9: Adding resources to a category

Finally, the templates and style sheets (see Figure 10) can all be edited from within the administrative interface, allowing libraries to highly customize the appearance of their dbWiz interface, including the ability to change colours, fonts, headers, logos, wording, and the location of information on the screen.



DISPLAY LEVEL
basic | intermediate | advanced

BASIC TEMPLATES

| TEMPLATE | ACTIVE | TRANSFER | SANDBOX | DESCRIPTION |
|---|---|---|---|---|
| basic_searchbox | | | | This is the search box that is displayed to the user when basic search is selected. |
| profile_dropdown | | | | This is a list of pre-made profiles that the user can choose to search |

ADVANCED TEMPLATES

| TEMPLATE | ACTIVE | TRANSFER | SANDBOX | DESCRIPTION |
|---|---|---|---|---|
| advanced_searchbox | | | | This is the search box that is displayed to the user when advanced search is selected. |
| main_search | | | | Controls the overall display search screen. This template calls the specific screen template to fil ... more |

Figure 10: Configuring templates

A development “sandbox” is also available, allowing local administrators to test any changes they have made before transferring them to their active version of dbWiz.

## Communication Strategy

As a development project that involves partners from across a large geographical area, we needed to ensure that we were maintaining effective communication from the beginning. We initially relied upon traditional methods such as a web site, an e-mail list, presentations at individual libraries, consultations, discussions at consortial library meetings, and conference presentations. Two new means of communication include the creation of a web-based video screencast, which allows partner libraries to get a look at the software in action, without requiring an on-site visit. The latest screencast can be viewed at http://theresearcher.ca

We also created a project wiki for dbWiz (see Appendix 1). This provides an interactive web site where we post project documents, gather user feedback, and have begun to create a wish list for a potential third version of dbWiz. The wish list has been very important in allowing us to remain open to innovations and new ideas without feeling the need to incorporate every suggestion, without delaying our committed release date. Unlike a traditional web site, which generally offers only one-way communication, the wiki provides a truly collaborative space for sharing ideas and comments. The wiki is open for anyone to create a free account and begin posting their own ideas or responding to comments.

## Implementation

Development of dbWiz will be continuing into the summer of 2005 based on feedback received from the partner libraries. Usability testing with our target audience of novice users will also be an important part of the refinement process. We will begin implementing the product in July, in anticipation of the beginning of a new academic year. In the fall of that same year, we will be releasing the source code under the GNU General Public License, allowing anyone to download, install, and modify their own copy of dbWiz. We are very interested in seeing dbWiz move beyond the original partners and outside of academic libraries. We believe strongly that the product will prove beneficial to the wider library community.

## Ongoing Challenges

One of the most significant challenges in developing a federated search tool is creating and maintaining a local system that needs to interoperate with so many diverse systems. For this reason, the importance of the National Information Standards Organization (2005) Metasearching Initiative cannot be understated. By bringing together the major stakeholders in federated searching, including libraries, content providers, and developers, NISO is facilitating the discussions that need to happen between these unique players with both common and divergent interests. Their work in standardizing access management, collection description, and search and retrieval processes will make federated searching easier, faster, and more efficient for everyone, whether operating in a commercial or an open source environment.

Another key challenge will be the ongoing maintenance of the search plugins for the different resources. As Hane (2003) mentions in "The truth about federated searching", if there are more than 100 resources being searched, and each one changes an average of two or three times a year, that still averages out to an update needed almost every day! We are currently developing an automated search system that will run overnight and report any errors discovered, allowing us to rapidly update any plugin that fails due to an unexpected change in a resource.

In addition to maintaining the existing resource plugins in dbWiz, we are also challenged by the number of resources that our partner libraries will continue to acquire and the need to create new plugins for all of these. In both cases, the maintenance of the project requires an ongoing commitment beyond the initial development of the software.

Robust relevance ranking is one of the key factors that put Google to the top of web searching. By crawling and indexing an ever larger part of the web, Google is able to apply their unique relevance algorithm to the collected data and produce highly relevant results. For federated searching however, creating our own index is not an option, due to our limited access to the vendor data. Again, efforts at standardization may help with this problem, but for the current version of dbWiz we will be applying some fairly simple keyword counting from the returned records to determine some basic relevance.

Deduplication is another issue all federated search products need to deal with. Due to the lack of standards in how the data in the different resources is structured, determining duplicates programmatically is very difficult. Also, as federated searching only downloads a portion of the search results (e.g. the first ten from each resource), deduping among the complete results is "virtually impossible" (Hane, 2003). We are anticipating some progress from the NISO Metasearch Initiative before we begin to explore the options for deduplication for any future versions of dbWiz.

## Project Significance

Despite these challenges, the benefits of developing dbWiz are significant. The ability to introduce our many Google users to the rich content in our libraries – and allow them to see for themselves what our subscription resources have to offer, cannot be over-estimated. Based on their experience with dbWiz, these novice users may even begin to try some of those resources directly, making it an important information literacy tool, allowing users to "learn by doing".

The dbWiz project also highlights the importance of library collaboration. A project of this scale would be too ambitious for any but the largest of libraries to attempt on their own. It would also be prohibitively expensive to maintain and keep up-to-date. Through working together, however, several medium and small academic libraries have been able to fund this project and see a successful, supported product emerge.

The dbWiz project also reveals the power of the open source development model. The tools used to build dbWiz, including Apache for the web server, mySQL and postgreSQL for the database, and Perl as the programming language are all open source and are among the most robust and stable available. If dbWiz is adopted by other institutions or consortia, there is opportunity for participatory development within the open source model – which further complements and enhances the collaboration mentioned above.

Finally, commercial federated searching products are beyond the budget of many medium or small academic or public libraries. Through the licensing of dbWiz under the GNU General Public License, these libraries will now have an entry point into the world of federated searching.

## Conclusion

Although it a very new product, dbWiz will soon become a mature component of the open source library environment. Based on a clear set of development objectives and secure funding from several partner institutions, dbWiz has moved from a proof of concept prototype to a fully-functional federated searching tool in just over a year. Its success reflects the benefits of federated searching, the open source model, and library collaboration.

## References

Abram, S. (2005), "The Google opportunity", *Library Journal*, Vol. 130 No.2, pp.34-5.

Hane, P. (2003), "The truth about federated searching", *Information Today*, Vol. 20 No.9, pp.24.

Luther, J. (2003), "Trumping Google?, Metasearching's promise", *Library Journal*, Vol. 128 No.16, pp.36-9.

National Information Standards Organization (2005), *Metasearching Initiative*, available at: http://www.niso.org/committees/MetaSearch-info.html (accessed 17 March 2005) .

Tennant, R. (2003), "The right solution: federated search tools", *Library Journal*, Vol. 128 No.11, pp.28-9.

Webster, P. (2004a), "Breaking down information silos: integrating online information", *Online*, Vol. 28 No.6, pp.30-4.

Webster, P. (2004b), "Metasearching in an academic environment", *Online*, Vol. 28 No.2, pp.20-3.

## Appendix - Additional Resources

Apache Software Foundation (2005), Apache, available at: www.apache.org/ (accessed 17 March 2005).

Free Software Foundation (2005), The GNU General Public License, available at: www.fsf.org/licensing/licenses/index_html#GPL (accessed 17 March 2005).

MySQL AB (2005), MySQL, available at: www.mysql.com/ (accessed 17 March 2005).

Perl Foundation (2005), The Perl Directory, available at: www.Perl.org/ (accessed 17 March 2005).

PostgreSQL Global Development Group (2005), PostgreSQL, available at: www.postgresql.org/ (accessed 17 March 2005).

Simon Fraser University Library (2005), dbWiz Project Wiki, available at: http://lib-cufts.lib.sfu.ca.proxy.lib.sfu.ca/twiki/bin/view/DbWiz/ (accessed 17 March 2005).

Simon Fraser University Library (2005), The reSearcher, available at: http://theresearcher.ca/ (accessed 17 March 2005).

Useful Utilities (2005), EZproxy by Useful Utilities, available at: www.usefulutilities.com/ (accessed 17 March 2005).

Wardley, A. (2004), Template Toolkit, available at: www.template-toolkit.org/ (accessed 17 March 2005).