

# Resource

## Pinpointing Your

BY ERIC JOHNSON  
AND NICK ANDREWS

WOULDN'T IT BE NICE  
IF THERE WERE A SYSTEM  
THAT WOULD JUST  
TELL US WHAT SOFTWARE  
HAS BEEN DEPLOYED  
ON EACH COMPUTER?

Our librarians at multiple service points needed a simple, fast tool to answer a common question: “Which computer has [pick your favorite] software?” Software packages are often expensive or require high-end machines to run, so not every software tool is available on every machine. Memorizing the location of each piece of specialized software is impractical, and written lists of software packages on each of hundreds of terminals are impractical and become rapidly outdated.

The solution was to create a webpage displaying an interactive map of our computer areas along with a list of available software. Hovering over a software title highlights each computer with that software. Clicking on a computer highlights every piece of software on that computer. Behind the scenes, we accomplished this by combining a spreadsheet of computer names correlated to the software on each machine with a map of computer locations. The same methodology can be used to answer other resources questions, such as, “Which study rooms have whiteboards?” and “What are the hours and services at each branch location?”

### Identifying In-Demand Software

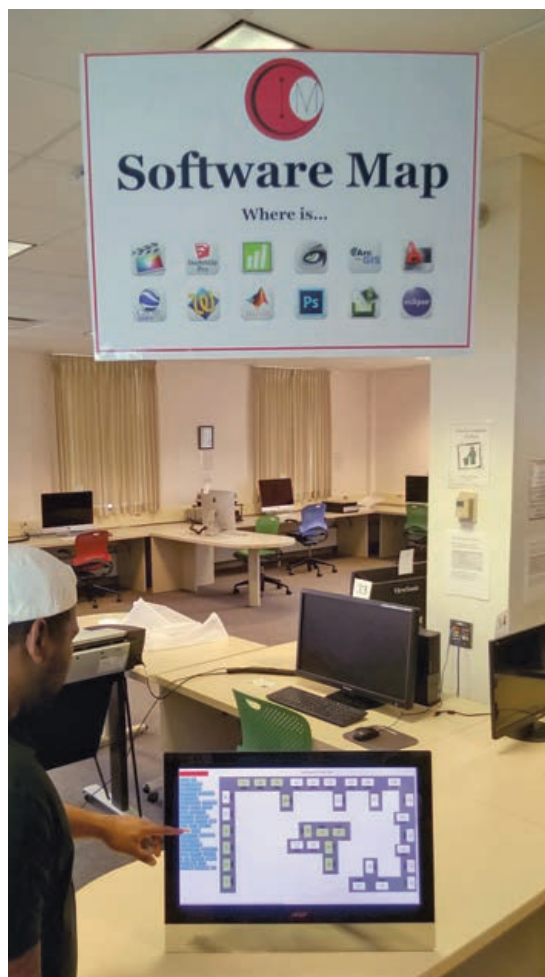
We started by asking what pieces of software users were interested in. Generating a list of software that users need is a difficult process to automate, since computers have thousands of programs. People actively use only a small number of these, but the entire list shows up on software reports. To create a comprehensive set of only relevant programs, we ended up looking at our Libstats record to discover what programs users requested directly ([code.google.com/p/libstats](http://code.google.com/p/libstats)). After

# Locator Maps: Library's Software Stuff

every patron interaction, library reference staffers use Libstats to create a brief, categorized report of what was asked, along with the answer that was given. We read through 2.5 years' worth of questions to get a list of software that users had asked about. We decided that for this project, "Where can I find X?" questions were more relevant than, "How do I use X?" questions. If no one had asked a question about a particular software title, it could mean several things. One is that the person knew where to find that software because it was on every computer—or he had already found a computer with it installed. The second possibility is that she didn't ever use that software. And a third is that he had never heard of that software, so he didn't know to ask about it.

For completeness and exposing people to new software, we also added selected additional titles to the list. And we included OSs so that people familiar with a particular user interface could find that same interface in our computers.

Wouldn't it be nice if there were a system that would just tell us what software has been deployed on each computer? Our IT department has



*Software map in use*

two systems that should have been able to give us that information. We use automated deployment of various com-

puter configurations. When software is updated, the IT people edit a profile of what software is installed on each computer and then push the updates through the network. However, we found a mismatch between what was reported to have been installed and what was actually on each computer.

Another resource tool is Labstats, which gives very detailed reports about how our computers are being used ([labstats.com](http://labstats.com)). We use an older version that tracks what programs are being used on each computer as well as when each computer is logged in to. While we can generate software usage reports, if a particular application hasn't been used on a given machine yet, the system won't report it. So with new deployments, a list of installed software is not available. However, we were able to augment our list of relevant software based on the actual hours each title was used campuswide.

## Creating the Database

After creating a list of in-demand software by looking at our Libstats records as described previously, we put these titles into a shared, online spreadsheet. Student workers logged in to each computer, verified the

Type of resource	Internal name	CIM06	CIM07	CIM13	mac06	mac07	mac12
54	Software Microsoft Word	1	1	1	1	1	1
55	Software Minitab	1	1	1	-	-	-
56	Software Movie Captioner	-	-	-	-	-	-
57	Software Mpeg Stremclip	-	-	-	-	-	-
58	Software Notepad++	1	1	1	-	-	-
59	Software Pages	-	-	-	-	-	-
60	Software Photobooth	-	-	-	1	1	1
61	Software Python	-	-	-	-	-	-
62	Software R	1	1	1	-	-	-
63	Software SAS	1	1	1	-	-	-
64	Software Sibaline	-	-	-	-	-	-

Spreadsheet to record software installed on each computer

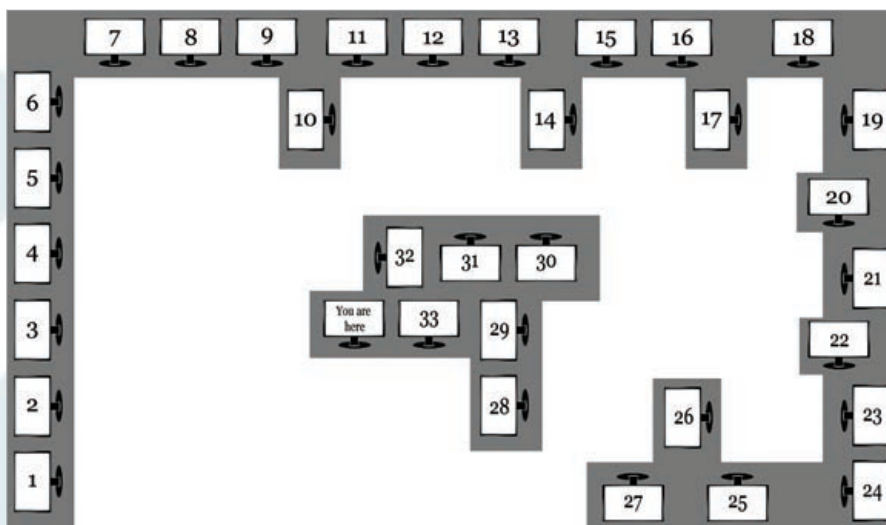
presence or absence of each software title, and recorded the result. If software was present, a “1” or the software version number was placed in the cell that matched the computer to the software title. If the software wasn’t found, a minus sign was placed in the cell. This way, we could quickly determine which computers hadn’t been surveyed. Since software deployments change, this is a task that we foresee doing in slack time before each new semester.

### Creating the Map

To turn our database into an interactive tool, we drew up simple maps of each computer area and edited them to become dynamic. We accomplished this by creating a copy of each map and mak-

ing transparent areas that represented each computer. This allowed us to create highlighted areas in HTML that would show up “sandwiched” between the original map on the bottom and the transparency layer on top. We used Adobe Photoshop, but there are many image-editing tools that will do the job. Our process was fairly simple; using the Photoshop Magic Wand tool to select contiguous areas of the same color, we deleted them to create the transparent areas needed. Then the images were saved as PNG files, because the PNG format handles transparent areas well and works in every browser.

The map was then brought into an HTML image-map editor to create highlight features that indicate where each piece of software can be found. Image



Basic map of the computer lab

maps are specific representations in web design that list points or areas on a picture. Areas are defined and then assigned an identification code. When the full map is displayed in a web browser, clicking an identified area can be programmed to produce various effects, such as linking to another webpage or redrawing the current webpage. For this project, we wanted a user selection to trigger highlighting of the selected computer or program, as well as the associated programs on that computer or the group of computers with that program.



### TO BRING IT ALL

## TOGETHER, WE NEEDED TO INCORPORATE OUR SOFTWARE LOCATION INFORMATION.

There are several free or inexpensive online image-map editors available. We used an advertising supported program called Online Image Map Editor, which also has a paid, downloadable version ([maschek.hu/imagemap/imgmap](http://maschek.hu/imagemap/imgmap)). We uploaded our map and drew squares to identify the areas that we wanted highlighted.

Since the highlight will occur below the top layer of our map, the areas didn’t need to be perfectly aligned to the transparent areas created earlier. They just needed to include the entire section to be highlighted without including any adjacent highlight areas. The Online Image Map Editor website then created an HTML code snippet that defined each area to be highlighted.

Our new HTML code snippet was then placed into a word processor. Word’s “search and replace” feature was used to add tabs between each area by searching for “/><area shape” and replacing it with “/>^t<area shape”. The new text was then copied into a spreadsheet, where the tabs we inserted caused the code referring to each “area shape” to be placed in a separate column. Each column in the spreadsheet, therefore, now referred to a specific computer shown on our map.



## WE USED

**STUDENTS, DURING SLACK  
TIME BETWEEN SEMESTERS,  
TO DRAW UP THE MAP  
AND FILL OUT THE  
RESOURCE SPREADSHEET.**

To bring it all together, we needed to incorporate our software location information. We added to our list of computers and associated software a row for the HTML image-map information. Then we added a set of cells with parameters such as highlight color, spacing, and page heading. A macro linked to a button in the spreadsheet then generated HTML code and placed it in the clipboard. Then

we could simply paste the code into a new text file named "index.htm" and place it on the web server, along with supporting files such as the JavaScript, images, and CSS files.

## Bug Fixing

When a user selects a piece of software, the software needs to be highlighted along with the corresponding computers. Likewise, when a computer is selected, it needs to be highlighted along with corresponding software titles. One of the problems we faced is that when a computer area triggered a software button to highlight, that button would then trigger highlighting of more computers until everything was highlighted. It was a cascading situation in which a button triggered an area to be selected, and that selection triggered more buttons, and so on, until every button and area was highlighted.

To solve this problem, we used two different selection methods: hover for buttons and click for areas. The buttons are triggered by hovering over them rather than by a click selection. Alternatively, while hovering over a computer highlights that computer aiding the user in pointing, the JavaScript that highlights the software buttons is not triggered until the computer area is actually clicked. Touchscreens normally use different input methods from mouse-based interfaces, but this split selection method allows the site to work on tablets as well as desktop computers.

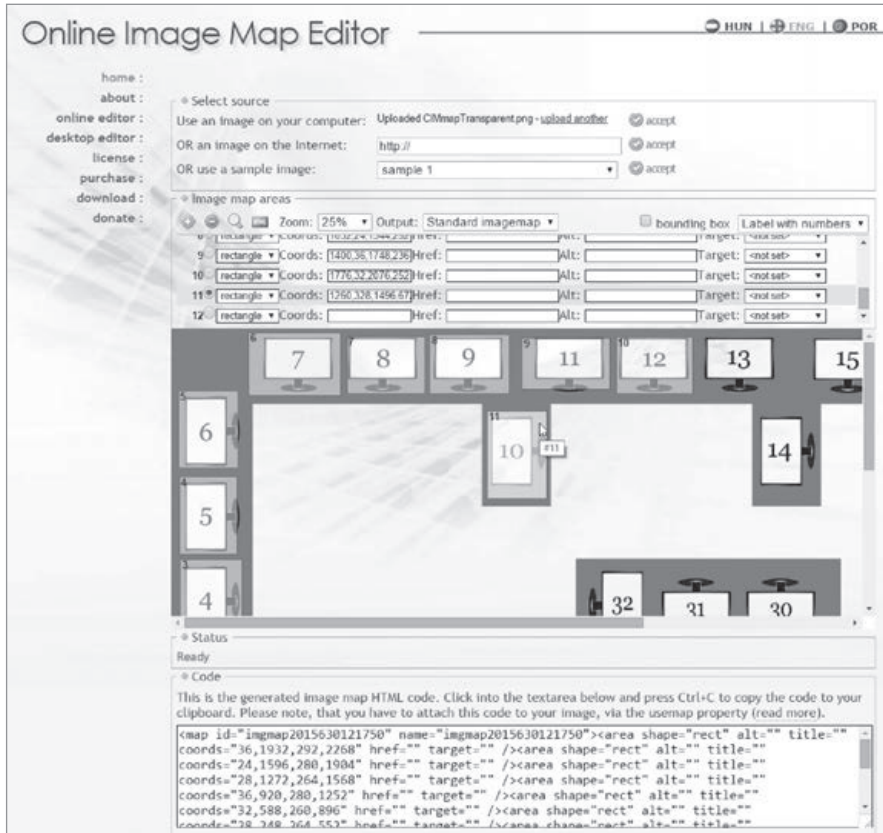


Image-map editor and resulting HTML

	A	B	C	D	E	F	G
1		paste imagemap here->		<map id="imgmap20151231532:<area sha<area sha<area sha<car			
2		paste softwareinventory ->	Library	King	King	King	I
3	page title =	Software in the CIM	Floor	1	1	1	
4	background color=	PaleTurquoise	Room or area	CIM	CIM	CIM	
5	Zone or Area=	CIM	Name for patrons	1	2	3	
6	map image file =	CIMmapTransparent.png	I.T. name	CIM14	CIM02	CIM03	C
7	map image width=	3999	Software	Windows	8	8	8
8	map image Height=	2333	Software	Mac			
9	Buttons area % =	17	Software	Adobe Digital Editions	1	1	1
10			Software	Adobe Dreamweaver	-	1	1
11			Software	Adobe Illustrator	-	1	1
12			Software	Adobe InDesign	-	1	1
13			Software	Adobe Photoshop	-	1	1
14			Software	Adobe Premiere Pro	-	1	1
15			Software	ArGIS/ArcMap	1	1	1
16			Software	Audacity	1	1	1
17			Software	AutoCAD	-	-	-
18			Software	Camtasia	1	1	1
19			Software	ChemDraw	-	-	-

Spreadsheet to generate HTML

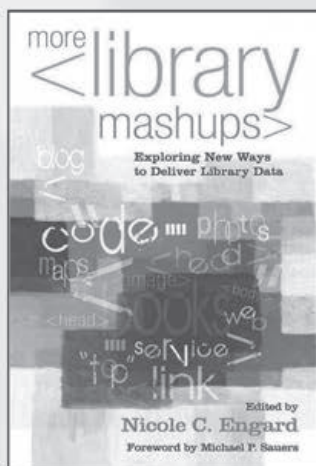
## The Final Product

Except for programming, the cost for this project was minimal. We used students, during slack time between semesters, to draw up the map and fill out the resource spreadsheet. After the template was created, it didn't take long to create the image map and produce the HTML code.

## What We Learned

Automatically generating a list of installed software would have been nice, but we found that it took actually looking at each machine to be sure what software was available on it. The process of sitting down and logging in to each machine

## A NEW RESOURCE FOR CREATING CUTTING-EDGE LIBRARY SERVICES



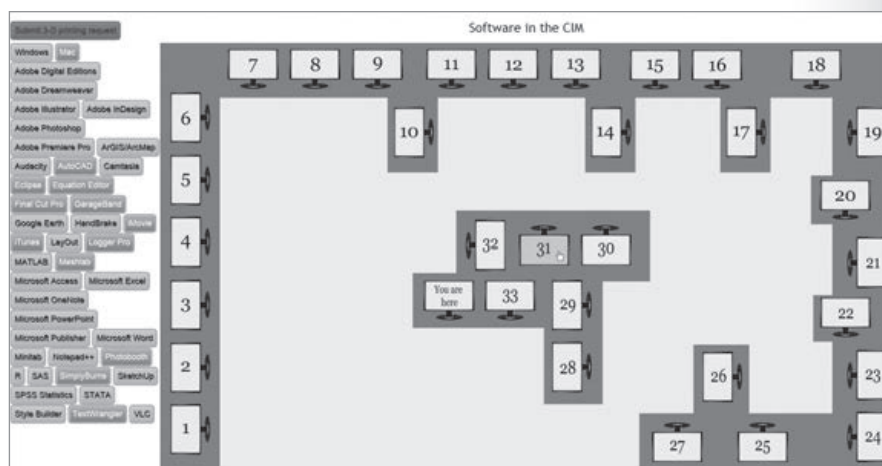
Edited by Nicole C. Engard  
Foreword by Michael P. Sauers  
ISBN 978-1-57387-498-4/\$45

“There’s something useful [here] for all types of librarians, from reference to systems to catalogers. Whether you code or not, you’ll find great ideas throughout.”

— Jason Puckett, author  
*Zotero: A Guide for Librarians, Researchers and Educators*

Available wherever books and ebooks are sold, or call  
**(800) 300-9868 or  
(609) 654-6266.**

**infoday.com**



Software resources map

and recording the information was tedious—so it had to be broken up into manageable chunks of time and assigned to the proper student.

### Evaluation

Anecdotally, the system gets plenty of use. In the Center for Information Management (CIM), the display is located near an information desk so additional questions can be answered if needed. Staffers like how easy it makes finding which computer to send people to if they had a specific task. Its presence reduces staff time answering “Where do I find ... ?” questions so they can focus on “How do I use or do ... ?” questions. Librarians staffing the main library information desk use the maps to quickly direct students with specific needs. Additionally, we will add a click-counting code to a future version that will let us know which software and what computer locations are of most interest to users.

### For Next Time

We have two computer areas online, and we are now mapping the remaining areas. All the software maps will be integrated into our InfoScreen system—displays in our library entrances that showcase campus and library maps along with notices of daily events. We will also redraw our maps. Now that we have some design samples, we are developing a standard set of icons, coloring, and visual design elements to guide users. We will also be adding items such as

study rooms, recording equipment, and printing stations to our list of searchable resources. Since users don’t always know the names of relevant programs, we will also look at creating categories of software so that they can select a type of activity, such as editing videos, and find a list of computers and software that can help them accomplish their task.

## RESOURCES

- Our free spreadsheet that generates HTML code and instructions for its use are available ([hdl.handle.net/2374.MIA/5260](http://hdl.handle.net/2374.MIA/5260)), along with the CSS and JavaScript codes
- Credits—[rwdImageMaps.js](http://rwdImageMaps.js) for resizable image-maps code ([github.com/stowball/jquery-rwdImageMaps](https://github.com/stowball/jquery-rwdImageMaps))
- Some of the code from [MapHighlight](http://MapHighlight) ([davidlynch.org/projects/maphilight/docs](http://davidlynch.org/projects/maphilight/docs))
- CSS button generator ([css-button-generator.com](http://css-button-generator.com))

### About the Authors

**Eric Johnson** is the numeric and spatial data services librarian in the Center for Digital Scholarship at Miami University.

**Nick Andrews** is the library technician in the Center for Information Management at Miami University.

Copyright of Computers in Libraries is the property of Information Today Inc. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.