8-2014

Yale Library IT News, July/August 2014

Yale Library IT Staff

Yale University Library IT News
July/August 2014

Library IT News:

- Michael Dula's office hour this month will be on **Thursday, July 31^(nd) from 11:00-12:00**.

- **Jenn Nolte** has accepted the position of Emerging Digital Services Analyst in Library IT. This is a new position that is the result of a reorganization in the department.   In her new position Jenn will research and evaluate emerging technologies, services and partnerships that together will contribute to the increased effectiveness and usability of the library for patrons and staff. A crucial part of Jenn's work will be to empower library staff to incorporate new library technology into instruction and outreach. To this end she will create training and documentation for staff in their use of discovery systems, content management (YaleSites, Libguides), and collaboration systems (Sharepoint). She will explore opportunities for innovative integration between these and other library, university and partner systems.

  Please congratulate Jenn on her new role in Library IT.

- Staff departures

  Andrew Predmore will leave the library on July 31 to take a position as a Senior Developer at Blink Reaction, which is a web development company. Andrew has been with the library as its Manager, Web Operations for close to three years. During that time he has implemented a new web design and brought the library into alignment with the rest of the University by taking us from our own standalone web content management system to the ITS-supported YaleSites/Drupal service. This accomplishment alone has brought the library to a more robust environment for support of web services going forward. In addition to many other accomplishments for our web site, Andrew implemented a new look and feel for Yale's Libguides, creating a much more polished presentation for this material. Most significantly, Andrew has tirelessly mentored staff and has been very active teaching Drupal, both within the library and in the larger Yale community, and his efforts have resulted in much more knowledge about web content management and Drupal than we've had before.

  Although Andrew will be missed in the library, please congratulate him on his new position and in his new endeavors.

  Julie Niemeyer has accepted a position in the Philosophy Department at Yale. Her new job title is Sr. Administrative Assistant, Faculty Support. She will be the assistant to the

department Chair and the Registrar, responsible for providing a high level of academic administrative services in support of faculty and graduate students. This will include coordinating teaching fellow assignments, recruitment, admissions, course listings, degree requirements, placement, faculty appointments and promotions. She will also maintain the Philosophy Department's website in Drupal 7. Her last day working in the library will be August 6th.

Julie will be missed in Library IT, but she will not be far away!

IT Project updates:

- **Lenovo:** Yale ITS recently announced a new university-wide agreement with Lenovo for standard personal computers at Yale. The Lenovo agreement replaces a previous agreement with Dell. The new Yale standard Lenovo PC pricing is 20-25% lower than the Yale standard Dell PC, and the delivery times will now be days instead of weeks for any Yale standard model. The Workstation & Technology Services unit is currently testing our installation and configuration procedures with the new Lenovo systems. The Sterling Memorial Library Nave will be equipped with a combination of new Lenovo all-in-one systems and mainstream desktops in late August.

- **Aeon:** Library IT was successful with a contract re-negotiation for Aeon, provided by the vendor Atlas. Aeon is a special collections circulation and workflow management system, currently used by Beinecke and Manuscripts and Archives. All of our requests were met during renegotiation, including a single invoice for all of YUL, a new renewal date of July 1 to coincide with our Fiscal Year, a single signature for the annual renewal, and the biggest news is an annual savings of $23,000.00. Over the next 5 years, YUL will save almost $120,000 in licensing and implementation costs for Aeon. We are now in a position to discuss additional special collection units and Yale partners implementing Aeon without the constraints of cost and reaching a site license cap.

- **Ares:** The Ares server was upgraded to version 4.3 and several system enhancements were also incorporated into the rollout, including batch activation and deactivation of course reserves from Ares into Voyager. This locally developed program will save no less than 6 full weeks of staff processing time annually. This was also a successful rollout of the Ares staff client in coordination with University ITS' EndPoint Engineering group.

- **Digital collections:** Several items have been added to the Digital Collections page (http://findit.library.yale.edu/) including the Israel Sack Furniture Archive and several more Sanborn Fire Insurance Maps. The number of Day Missions items has also increased significantly. The following screen shot of the facets on the Digital collection page shows the total number of items ingested into Blacklight in each collection.

Limit your search

Creator	>
Date	>
Content Type	>
Topic	>
Digital Collection	∨

Day Missions Periodicals	3045
Day Missions Annual Reports	2171
Yale Indian Papers	601
Lewis Walpole Library	500
Sanborn Fire Insurance Maps	404
Maurice Durand Han Nom	187
Arabic and Persian Medical Books and Manuscripts	62
Persian Philological Texts	22
Israel Sack Furniture Archive	17

Features:

- **Collaborative software survey results – by Katie Bauer**
  In June the Web Strategy Group started to work with Greg Blasko to determine how we can better help staff who need to use software to collaborate on projects or for department workflows. As a first step we decided to ask people to tell us what software they use and how they use it. We designed a quick Qualtrics survey tool and sent it to staff via Yulib email. 82 people took the survey. The following summarizes questions and answers.

  Frequency of Use
  The most commonly used tool for collaboration was email (76 staff reported using email attachments daily or at least once a week), followed by public library web sites (68) and SharePoint (52). Every one of the thirteen digital tools listed was used by a respondent and other tools were mentioned that we didn't list, such as Sakai (the course management tool).

  Type of Use
  We asked about categories of use, and the types of tools used in those categories.
  The most cited collaboration tool was email, which has no version control or simple way to merge changes from collaborators. Email and Box were often used in all categories. The table below shows the top five tools listed for each type of use.

	Sharing Internal Library Information	Posting Public Facing Information	Collaboration or document version control
1	SharePoint (69)	Libguides (36)	Email attachments (52)
2	Email attachments (68)	YUL Public Web (28)	SharePoint (44)
3	Staff Web Site (49)	Email attachments(16))	Box (43)
4	Box (39)	Staff Web Site (8)	Google Drive (32)
5	YUL Public Web Site (34)	Box (7)	Basecamp (26)

Ease of Use
Email attachments were rated the easiest to use and SharePoint the hardest. 43% found SharePoint difficult to use, and using SharePoint frequently did not improve that much: 33% of frequent SharePoint users still found it difficult to use.

Satisfaction
Respondents were most satisfied with email (average rating of 2.75 out of 3) and least satisfied with SharePoint (1.69 average rating).

Comments
One theme emerging in comments was that the number of software tools in use across the library can be frustrating. For example, "The biggest problem I have is that every group/committee/project seems to use a different system for versioning."

Clearly SharePoint was on the minds of lots of people who answered the survey. Out of 26 comments 16 were about SharePoint. Some departments have adopted it for their work, but they also tend to find it frustrating to use.

Outcomes
Given how much SharePoint is used in the Library, and its low rankings for ease of use and satisfaction, we have concluded that we can best help the library by concentrating on a SharePoint 365 implementation with some improvements in implementation and support. Greg Blasko is currently working on these plans. In addition, to try to help simplify the environment of collaboration tools in use at the Library, the Web Strategy Group will develop a list of recommended tools for specific situations. Look for news about these efforts soon.

See full results of the survey.

- **LadyBird/Hydra performance:** a brief statistical wrap up - by Mike Friscia
  - Objects added to Ladybird between July 1 and now: **933,306.**
  - Objects ingested into Hydra since the launch in summer 2013: **496,160.**

The more interesting statistics have to do with the movement of items into the systems. In July 2013 the average time it took to import an object with metadata into Ladybird was 27 seconds and the average time to ingest an object into Hydra from Ladybird was 17 seconds. Updates to metadata only would take about 2 seconds in each system. These rates seemed perfectly fine until we began to consider the scale of Kissinger. Simply put, if we did not speed things up, we would fall behind about two days every two weeks which compounded over a year could mean the ingest operations could lag behind by as much as 100 days.

Two approaches were used to speed up ingest. The first was putting the applications on better hardware. So the Systems Infrastructure team configured and installed new hardware to host the applications this past winter which gave an immediate boost to the performance of all the applications. The next step was to go through the software and identify ways to make them perform various operations quicker. Then you analyze the application to see which parts are independent of other operations and design the application so that more than one thing is happening at a time. The two key goals are creating an application that uses parallel processing and multithreading. Parallel processing means that two or more events related to a single operation can take place at the same time. Multithreading is making it so that more than one operation can take place at a time.

So a simple analogy is this, you are cooking dinner and plan to make a roasted chicken and some boiled carrots. From a programing standpoint, this is how you would do this the first time. You would turn on the oven and wait for it to reach 350 degrees. Then you would put the chicken in a pan and then into the oven. Wait an hour or so and then take the chicken out. Next you start to chop carrots and then you fill a pot with water, then add the carrots, turn the stove on and wait until they are done. This is how to approach it to make sure that each procedure was performed from start to finish. By focusing only on one thing at a time from start to finish, success is almost guaranteed but terribly inefficient since this probably took a couple hours since no task could begin until the current task was 100% complete. So we introduce the parallel processing. The second time around we realize that the cooking of the chicken and carrots can happen kind of at the same time, neither has any steps that rely on one or the other being even partially completed. So we preheat the oven and start chopping carrots, then put the chicken in the oven and while that's cooking, put the carrots in the pot and start boiling them. This is much better, we were a little busier during the process but instead of taking a couple hours, we're done in about an hour. Finally we decide that instead of cooking one chicken, we are going to cook a hundred of them and there's a catch, we do not want to alter the process at all. So working quicker is not an option and we only want to be responsible for one chicken and one pot of carrots. The only solution is that instead of one person, we will have one hundred people and everyone gets their own kitchen. So if everyone starts at the exact same time, they should all end at the exact same time.

So programming the ingest operations for Ladybird and Hydra basically takes the same path. As a result the boosts to performance on the hardware and making the applications for Ladybird import more efficient we reduced the time it takes to import each object from 27 seconds to about 13. But the more interesting thing is that we can process 25 at a time which effectively means we can import 6,428 objects per hour instead of 133. So the end result is that we now have software that can ingest very large collections with little impact to other digital collection work taking place.