FINAL REPORT FOR IMLS GRANT NO. LG-46-11-0078-11

Authors: Amy Rudersdorf, Dean Farrell, Lisa Gregory, Kathleen Kenney
Digital Information Management Program, State Library of North Carolina

ADMINISTRATIVE INFORMATION

Institution: State Library of North Carolina (Department of Cultural Resources)
Project Title: CINCH: Capture, Ingest, & Checksum tool
Award Amount: $25,000
Grant Period: August 2011 – August 2012
Project Director: Amy Rudersdorf
Project Partners:
  o State Archives of North Carolina (Department of Cultural Resources)
  o Belk Library, Elon University
  o J. Murrey Atkins Library, University of North Carolina at Charlotte
  o North Carolina Libraries for Virtual Education (NC LIVE)

PROJECT SUMMARY

Many large institutions have developed digital curation tools and micro-processing applications to manage digital heritage and data resources over the long-term. These tools and applications often have great potential for use by the wider library and archives community. Unfortunately, they are often produced to respond to the unique programmatic needs of an institution. Additionally, even lightweight, open-source tools can require a robust localized infrastructure and significant human and computing resources to implement and administer. While this work is invaluable to the larger community in furthering knowledge about the functional possibilities of digital curation tools, the applications themselves simply may not be easily transferable to smaller or less-well funded institutions.

An argument can be made that institutions without well-established infrastructures should not take on the responsibilities of digital preservation because their efforts may not be sustainable. This may be true – in theory. In practice, however, small and mid-sized institutions have similar digital archiving mandates to larger institutions, from recording the academic history of their college to long-term preservation of state and local records. Although they may not share the technical infrastructure or human resources to maintain complex systems that larger institutions boast, the mandate remains and tools to support these efforts are necessary. For this reason, the State Library set out to develop the Capture, Ingest, and Checksum tool (CINCH) to be portable and flexible enough for use by smaller institutions. Briefly, CINCH does the following:

- Grabs freely available online content
- Makes sure it stays unchanged while being processed
- Extracts limited metadata and,
- Uses digital preservation best practices as it readies it for deposit into a repository.

It is our hope that CINCH will also lessen the burden on repository managers in at least two ways. First, it provides repository managers with a ready-made digital package that contains technical,
administrative, and descriptive information to support long-term preservation of the content. This package (otherwise known as a Submission Information Package, or SIP) is ready for whatever local preservation actions an institution undertakes. Second, and most importantly, it will minimize the need to "hunt and peck" for content or badger content creators to submit their content, only to have to contend with various methods for deposit like secondary storage devices or email attachments. For example, many universities have schedules of retention requiring their faculty and staff to submit certain documents to the archives on a regular basis. Repository managers with limited staff find themselves dealing with numerous contributors, who have many different levels of technological expertise, and who have a variety of other activities ranking in importance (to them) above compliance with this type of mandate. Repository managers have found that encouraging voluntary submission of these types of materials is only weakly effective, even with a mandate for compliance. In a situation like this, repository managers could use CINCH to acquire a good portion of that content from a location that may already be part of the contributors’ workflow – the internet. In this way, CINCH moves beyond this traditional model and takes an active role in acquiring identified and appropriate content for digital preservation.

**PROCESS**

Administratively, few changes took place for this project. A server was leased to accommodate the software. While the lead developer is a current member of the library’s staff, a contractor was hired to provide assistance and consultation as needed during development. No organizational or policy changes were made in developing the project, although use of the tool may lead to such changes in the future.

The initial project requirements were laid out in early 2011 and can be seen below.

*Illustration 1. CINCH workflow.*
The State Library is legislatively mandated to collect, preserve, and provide access to North Carolina state publications, regardless of their format. For the Library's purposes, publications are documents created by state government for public consumption. In order to determine CINCH’s technical requirements, testing was first done to determine the number of files within a targeted web space that were in need of preservation under the Library's collecting scope. Files in .pdf format were isolated as highest priority.

The subscription-based Archive-It service, which is already used by the Library for its web archiving program, produces a report listing the URLs of .pdf files found during its crawls. Because the Library crawls North Carolina's state government web presence, those .pdfs are possibly within the Library's publications collecting scope. One of these reports was examined, and the URLs were winnowed down by removing those with words such as “calendar” and “meeting” in them, as most likely pointing to files that the Library does not consider publications. This still left well over 100,000 files to collect and preserve using CINCH.

The technical workflow followed out of the micro-services approach developed by the California Digital Library whereby elements could easily be added or removed as necessary to further the project.

There was also some initial debate over what programming language and platform might best suit the application given potential resources and current programming expertise. Node.js, Ruby on
Rails, and PHP were considered. Node.js allows very high throughput for such things as file downloads, but was ultimately rejected due to it still being a relatively nascent project that had no stable release. Ruby on Rails was primarily looked at due to developer interest and its use in other large scale library/archival related digital preservation projects such as Hydra developed by the AIMS project and DAITSS developed by the Florida Center for Library Automation. It was rejected for several reasons: it was unclear if the developers could get up to speed quickly enough to use it effectively during the grant cycle, and it was decided that maintaining development language consistency with other in-house projects was important for an organization as small as ours in which hiring/Replacing developers is a non-trivial process.

There were several mid-course technical corrections. First, the project initially aimed to be easily usable by Windows and Linux operating systems. However, it was difficult to find a suitable open source anti-virus tool that worked well on Windows. The result is that the current iteration of the CINCH tool is only compatible with Unix/Linux systems.

Secondly, a backend messaging system was tested that would have increased the throughput of file processing. However, the project’s programming language of choice doesn’t have a well-documented, reliable library to interface with the messaging system. If the project were started from scratch PHP probably would not have been chosen as the primary development language due to its lack of parallel processing functionality. However, adding a messaging system and development in an alternate programming language would also have added a great deal of complexity to the application setup, as PHP is easy to install and is ubiquitous on shared hosting services. These changes would have somewhat negated one of the project’s goals of providing a simple, lightweight tool.

A beta group, including internal library and archives staff as well as our project partners at Elon University’s Belk Library and J. Murray Atkins Library at the University of North Carolina at Charlotte tested the application. They went through each of the actions a CINCH end user would need to perform to successfully use the software. The were then asked to fill out a short web-based survey in which they were asked the following:

1. How easy was it to log in to CINCH?
2. How quick was it to find the documentation for CINCH?
3. How user-friendly is CINCH’s interface?
4. How useful is the included documentation for our software?
5. How successful is CINCH in performing the processes to package your born-digital content for ingest?
6. How often does CINCH freeze or crash?
7. Overall, are you satisfied with the performance of our software, neither satisfied nor dissatisfied with it, or dissatisfied with it?
8. How likely are you to recommend CINCH to others?
9. How can we improve the CINCH interface, documentation, or package output?
Based on their responses several bugs were uncovered and fixed. Secondly, some of the CINCH website verbiage was changed to take into account end users who might not be familiar with technical verbiage.

**PROJECT RESULTS**

CINCH does not relate directly to storage or access, but it does address the need for appropriately formatted SIPs to ensure the fixity, authenticity, and long-term manageability of digital assets. As a broadly accessible tool with generalized purpose, its greatest impact will be on small and mid-sized institutions and organizations that do not now have access to an application of this nature.

Two key advantages of the tool are that it (1) is free and relatively easy to implement, and (2) allows repository managers to meet content creators where they work by collecting their digital assets from agreed upon locations that can be part of the creator’s existing workflows. Its ease and minimal resource impact represents the best-case scenario for many institutions: a win for content creators, a win for digital repository managers, and a win for those who will need to access the content in the future.

CINCH 2.0 planning is already underway. Improving the quality and range of metadata and expanding the types of files beyond .pdf are at the top of the development list. Tools such as Apache’s Tika might be utilized to enhance metadata results and include full-text extraction, while OpenNLP or a similar toolset could be integrated to test natural language processing to try to extract other metadata based on its location within the file. For example, “title,” “author,” and even a document abstract, might be extrapolated from the title page or start of the document. And, instead of mapping keyword and subject term creation to the document’s author, attempting to create more robust description by extracting the most frequently used terms and phrases within the document (after filtering out stop words) is worth exploring. The tool may be ported to a different programming language (Ruby, Groovy, Python) to allow for easier integration with back-end messaging systems, parallel processing of files, and natural language processing integration. A final option, of interest to our partners, is to integrate a test for digital rights management (DRM) restrictions.

Even with a limited staff and technical resources it is very possible for small players in the cultural heritage arena to make useful technical achievements in the field. It does take a certain level of persistence and creativity to work through the inevitable staffing and technical restrictions that any small institution faces, but CINCH is proof that it can be done.
REFERENCES

http://cinch.nclive.org/
http://digitalpreservation.ncdcr.gov/cinch/index.html
https://github.com/SLNC-DIMP/Cinch
http://slnc-dimp.github.com/Cinch/


Rudersdorf, Amy. "Digital Preservation Ingest can be a CINCH." LibraryHiTech. (July 2012).