

This is a preprint of an article published in the [Russian Digital Libraries Journal](#), vol. 17,
no. 2.

A Model for Integrating the Publication and Preservation of Journal Articles

Kevin S. Hawkins
University of Michigan, Ann Arbor

There are policy, technical, and workflow gaps in library efforts to preserve online journal literature. Since libraries are increasingly involved in journal publishing, HathiTrust, a shared preservation-quality digital repository, is a natural place to archive and provide access to journal literature to ensure its long-term preservation and discoverability. The University of Michigan Library is funding the creation of mPach, an open-source, end-to-end publishing system in which archiving in HathiTrust happens as a byproduct of publication rather than being carried out after the fact. The architecture of mPach, its envisioned workflow, and plans for creating a shared infrastructure for publishing open-access journals are all summarized.

The deficit in journal preservation

Until quite recently, publishers produced documents on physical media, and libraries acquired and preserved copies of these documents. But in the era of the Internet, when publishers host content online, the library's role in acquiring and preserving the content is in jeopardy: without special licensing arrangements such as those often provided by open-access journals, a library has no legal right to make a copy of the content for preservation.

Various business models have evolved to address this situation, especially for journals, which are increasingly available only online. For non-open-access journals, research libraries often negotiate the right to create a digital copy of any content acquired during the period of subscription[1] and make this content available only to their patrons,[2] though few are equipped to provide this kind of restricted access and archiving with integrated browse and search functions. To address the more pressing concern of publishers going out of business without *any* libraries holding a copy of the content, libraries and publishers have collaborated in initiatives like LOCKSS,[3] CLOCKSS,[4] and Portico[5] in order to guarantee that one or more copy of the content will become available if it is no longer available from the publisher. Similarly, the Koninklijke Bibliotheek and Elsevier reached an agreement in 2002 whereby the KB will preserve Elsevier journals under terms similar to those governing journals that use LOCKSS, CLOCKSS, and Portico.[6] Still, there are problems with these models. LOCKSS and CLOCKSS use web crawling, which captures only the appearance of webpages but not their underlying structure or search functionality. Portico and the KB, on the other hand, rely on publishers to deliver journal articles in valid file formats, and not just the version first published but also any corrected versions of these articles.

One way to ensure that a library always has access to the latest content is for the library to operate the very system used to publish the journal. A survey in 2010 of a cross-section of North American academic libraries found that, of 144 responding institutions, 43 offered "operational publishing services" to their

scholars at the institution.[7] Of these 43 institutions, most host publications using open-source software such as Open Journal Systems (OJS)[8] or DSpace,[9] while about a quarter use Digital Commons,[10] a hosted platform provided by bepress. OJS and Digital Commons are also the dominant publishing platforms according to a survey in spring 2013 by the Library Publishing Coalition.[11]

Unfortunately, all of these platforms deliver to users only those files (primarily PDF files) created and uploaded by a journal editor. Since the library is not in a position to control the software and workflows used to create these files, the library can only provide bitwise preservation of the files, severely hampering future migration of the content.

A higher standard for preservation

Since libraries are increasingly involved in journal publishing, HathiTrust,[12] a shared preservation-quality digital repository, is a natural place to archive and provide access to journal literature to ensure its long-term preservation and discoverability. HathiTrust already archives and provides access to reformatted library holdings, but the University of Michigan Library, a founding member of HathiTrust, sees an opportunity to use HathiTrust for publishing born-digital journals as well. To develop an infrastructure in support of low-cost university-based publishing that addresses the needs and values of both content creators and librarians, the U-M Library is funding the creation of mPach,[13] an open-source, end-to-end publishing system in which the act of publishing and the act of archiving are unified. In other words, archiving in HathiTrust happens as a byproduct of publication rather than being carried out after the fact. mPach leverages existing components of HathiTrust and available open-source software where appropriate.

Archiving is not as simple as saving a copy of a file produced by a journal editor, as OJS and institutional repositories generally do. Instead, the content needs to be stored in a format that allows digital preservation. PDF/A, a non-proprietary variant of the PDF family standardized as ISO 19005, is often suggested for such needs, but even a PDF/A file is poorly suited for use with screen readers for the visually impaired and for any non-paginated display, and is suboptimal even for searching and data mining.

Rather than preserving the paginated appearance of a document, the text of the article needs to be stored in a format that reflects its structure and semantics, with associated media in formats that can be preserved and rendered. mPach has developed a specification for journal articles that uses the Journal Article Tag Suite (JATS), an application of NISO Z39.96-2012,[14] for the text and stores this with high-quality versions of media objects and with a METS record containing structural and preservation metadata.

An overview of mPach

There are three major parts of mPach (see also figure 1), each of which includes components in various stages of development at the time of writing:

- **the peer review and editorial system:** what authors and reviewers interact with

- **Prepper:** what prepares the article for ingest into HathiTrust for archiving and publication
- **modified HathiTrust components:** various modifications to existing components of the HathiTrust environment to support born-digital journal articles

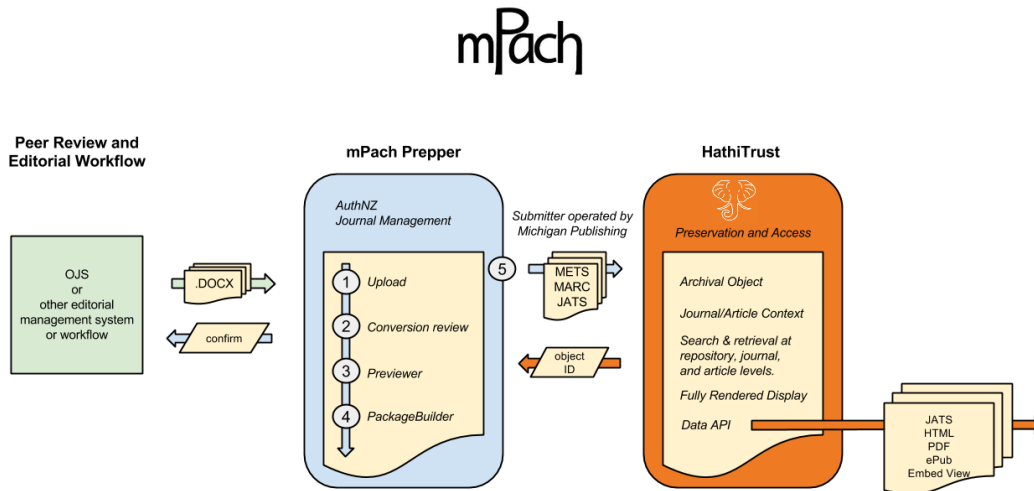


Figure 1: Major parts of mPach

As a modular system, mPach could be used with any peer review and editorial system that is capable of interacting with Prepper; however, the developers have chosen to provide OJS as the default option. Despite having no support for digital preservation, OJS is already widely used for library-based journal publishing, and mPach’s integration with this software will allow for a smooth transition of journals already published using OJS into the HathiTrust repository. Integration with mPach requires that manuscripts that reach the “layout” stage in OJS be sent to Prepper, which prepares the HathiTrust Submission Information Package (SIP).

Prepper provides a user interface for the editor of a journal: a dashboard for administering the journal and putting manuscripts through a production process—akin to composition and typesetting—that prepares all content according to the preservation standard developed for mPach content in HathiTrust. Prepper invokes Norm, a Python application developed to convert manuscripts from Office Open XML (“DOCX”) format[15] into XML that conforms to JATS. DOCX is the default option because, like OJS, it is widely used in the editorial process of journals published by libraries. The Prepper interface also guides the staff member through a review of validation errors detected by Norm’s conversion, uploading high-resolution figures, supplying “alt text” for figures, previewing the article as rendered using the default stylesheet (based on the Preview XSLT stylesheets[16]), uploading supplementary material,[17] and submitting for ingest into HathiTrust.

mPach requires a number of significant modifications to HathiTrust components and workflows originally designed to support reformatted print materials. The reading interface in HathiTrust, which previously supported only

display of digitized page images, renders JATS XML in HTML and allows a user to download a dynamically generated PDF and EPUB, display metadata specific to articles (figure 2), and link to a special “collection” for the journal in HathiTrust’s Collections application[18] that allows for browsing volumes and issues of the journal (figure 3).

The screenshot displays the HathiTrust Digital Library interface. At the top, there is a navigation bar with links for Home, About, Collections, Help, and Feedback. Below this is the HathiTrust logo and a search bar with options for FULL-TEXT and CATALOG. A search bar contains the text "Search words about or within the items" and a "LOG IN" button. Below the search bar, there are links for "Advanced full-text search" and "Search tips", and a checkbox for "Full view only".

The main content area is divided into several sections:

- Journal Information:** "Back to Journal of Electronic Publishing collection", "JEP the journal of electronic publishing", "Journal of Electronic Publishing", "Vol. 15, No. 1 (Summer 2012)", "About this journal", "About this Article", "Refurbishing the Camelot of Scholarship: How to Improve the Digital Contribution of the PDF Research Article", "John Willinsky, Alex Garnett, Angela Pan Wong", "View full catalog record", "Copyright: [CC] BY".
- Get this Article:** "Download (PDF)", "Download (XML)", "Download (EPUB)".
- Supplemental Materials:** "Data Set (XLS, 35K)".
- Add to Collection:** "Login to make your personal collections permanent", "Select Collection", "Add".
- Share:** "Permanent link to this article", "http://dx.doi.org/0000.0000.000".
- Version:** "2012-07-19 16:37 UTC".

The article content itself is titled "Refurbishing the Camelot of Scholarship: How to Improve the Digital Contribution of the PDF Research Article" by John Willinsky, Alex Garnett, and Angela Pan Wong. It includes a DOI link, a permissions icon, and a note that the paper was refereed by the Journal of Electronic Publishing's peer reviewers. The article is structured with an Abstract, Introduction, and a main body of text.

Abstract: *The Portable Document Format (PDF) has become the standard and preferred form for the digital edition of scholarly journal articles. Originally created as a solution to the need to “view and print anywhere,” this technology has steadily evolved since the 1990s. However, its current use among scholarly publishers has been largely restricted to making research articles print-ready, and this greatly limits the potential capacity of the PDF research article to form a greater part of a digital knowledge ecology. While this article considers historical issues of design and format in scholarly publishing, it also takes a very practical approach, providing demonstrations and examples to assist publishers and scholars in finding greater scholarly value in the way the PDF is used for journal articles. This involves but is not limited to graphic design and bibliographic linking, the deployment of metadata and research data, and the ability to combine elements of improved machine and human readability.*

Introduction: The Portable Document Format (PDF) was released by Adobe Systems in 1993 to facilitate the electronic distribution of documents. It was created to assist the circulation of digital documents among the newly networked computers that were spreading through offices, whether in local area networks (LAN) or through the Internet. What had become apparent was that documents were being prepared by various word-processing programs, each with their own proprietary file format. With networking racing ahead of file compatibility, John Warnock, Adobe Systems cofounder, in 1991 initiated what he called the Camelot Project in order to solve the “view and print anywhere” problem, as he neatly characterized it (1991, p. 1). Nearly a decade earlier, in 1982, the resourceful Warnock, working with Charles Geschke, figured they had solved the same problem with PostScript (marking the beginning of Adobe Systems). However, PostScript was itself not proving universally applicable. It required “powerful desktop machines,” as Warnock put it, as well as PostScript printers (1991, pp. 1–2).

The goal of Camelot was to develop a lightweight file format that would serve the broadest possible range of users, at least until widespread computing power caught up with the demands of PostScript. Camelot was intended, then, as a temporary, transitional solution to the view-and-print-anywhere problem. Its history and success proved otherwise. When launched in 1993, the file format’s poetic Camelot moniker was replaced by the prosaic “portable document format,” now universally known as PDF. In 2008, Adobe released the PDF as an open standard for others to develop applications for writing and reading it, in what we might think of as the new twenty-first-century corporate spirit of open standards and open source software.

In scholarly communication, the PDF has become the standard file format for research articles published in the electronic edition of peer-reviewed journals. Although many journals also publish a HTML version of their articles along with a PDF, the bulk of the research literature is now available in PDF. Over the last decade, the majority of researchers have switched to reading the online edition of journals available through their library’s electronic collections (King, Tenopir, Choemprayong, and Wu, 2009, p. 131; Hemminger, Lu, Vaughn, and Adams, 2007). While finding articles online is becoming a common practice, most academic faculty print out a good proportion of the PDFs they wish to read, while younger and more research-oriented scholars lead the way in reading articles on their computer

Figure 2: Mockup of an article viewed in HathiTrust’s user interface

The screenshot shows the HathiTrust Digital Library interface. At the top, there is a navigation bar with links for Home, About, Collections, Help, and Feedback. Below this is the HathiTrust logo and a search bar with a 'FULL-TEXT' and 'CATALOG' filter. A search bar contains the text 'Search words about or within the items' and a 'LOG IN' button. Below the search bar are links for 'Advanced full-text search', 'Search tips', and a 'Full view only' checkbox.

The main content area is titled 'Journal of Electronic Publishing' and features a search bar for the journal. Below the search bar are tabs for 'Articles (369)' and 'About This Journal'. A 'Sort by' dropdown menu is set to 'Date Descending'. The page lists several volumes of the journal, with Volume 16 (2013) at the top and Volume 11 (2008) at the bottom. Each volume is expanded to show its contents, including articles and books. For example, Volume 16 (2013) includes articles like 'The Short-Term Influence of Free Digital Versions of Books on Print Sales' by John Hilton, III; David Wiley, 'UP 2.0: Some Theses on the Future of Academic Publishing' by Phil Pochoda, 'Our Book' by Sandra Ordonez, 'Launching (and Sustaining) a Scholarly Journal of the Internet: The International Journal of Baudrillard Studies' by Gerry Coulter, 'Justify Just of Just Justify' by Mohamed Elyaaakoubi; Azzeddine Lazrek, and 'XML Production Workflows? Start with the Web' by John W. Maxwell; Meghan MacDonald; Travis Nicolson, et al. It also includes an 'Editor's Note' by Judith Axler Turner.

Figure 3: Mockup of a journal viewed in HathiTrust's user interface

Discovery of known items in HathiTrust using metadata like title and author is currently provided for by a catalog of MARC records, with one per item in the repository. For mPach, each article has its own analytic catalog record, tied to a monographic record for the journal as a whole. Finally, the HathiTrust Data API[19] allows for the content of each article to be retrieved for use outside of the native HathiTrust interface.

Note that by policy HathiTrust only closes access to content for legal reasons, not because a rightsholder wants to restrict access. Therefore, mPach only supports the publishing of open-access journals.

Workflow

In the typical workflow for publishing a journal using mPach, a journal editor uses OJS to manage submissions, peer review, and the editing process. Once an article reaches the “layout” stage (where a combination of composition and typesetting allows the article to be formatted in a consistent way), the journal editor formats it according to a predefined list of styles in Microsoft Word and submits the article in DOCX to mPach’s Prepper, which guides the editor through conversion to JATS XML, preparation of the SIP, and hands off to Submitter for ingest. Prepper keeps track of articles so that a revised version can be submitted for ingest. Currently the ingest process overwrites any previous version of an item with the same identifier, but eventually HathiTrust will archive past versions and allow users to navigate among them.

mPach as a shared infrastructure

In order to ensure only authorized deposit of content, Michigan Publishing, the primary academic publisher of the University of Michigan that is part of the U-M Library, will host the only instance of Submitter. Organizations wishing to publish journal literature in HathiTrust will be able to use Submitter either with their own instance of Prepper or with an instance of Prepper offered as a hosted service by Michigan Publishing. The developers envision extending the Norm component to handle OpenDocument (“ODT”)[20] and LaTeX as input formats, each of which is more commonly used in certain communities. Furthermore, now that the Book Interchange Tag Suite[21] has been adopted as a standard, the mPach architecture might be extended to support monograph publishing. While mPach is currently being developed to meet the needs of Michigan Publishing, the contribution of the sourcecode to the planned HathiTrust Development Environment should foster contributions from developers not at U-M and therefore lead to the creation of a truly shared infrastructure for publishing open-access scholarly journals.

References

- [1] Sadie L. Honey. Preservation of electronic scholarly publishing: an analysis of three approaches. *Portal: libraries and the academy*, 5(1):59-75, Jan. 2005.
- [2] NISO SERU Standing Committee. SERU: a shared electronic resource understanding: a recommended practice of the National Information Standards Organization. Baltimore: National Information Standards Organization. 2012. http://www.niso.org/publications/rp/RP-7-2012_SERU.pdf.
- [3] Lots of Copies Keeps Stuff Safe. <http://www.lockss.org/>.
- [4] CLOCKSS. <http://www.clockss.org/>.
- [5] Portico. <http://www.portico.org/>.
- [6] National Library of the Netherlands and Elsevier Science make digital preservation history: permanent digital archive assures perpetual accessibility of scientific heritage. August 20, 2002. <http://www.kb.nl/en/news/news-archive-2002/national-library-of-the-netherlands-and-elsevier-science-make-digital-preservation-history>.

- [7] James L. Mullins, Catherine Murray-Rust, Joyce L. Ogburn, Raym Crow, October Ivens, Allyson Mower, Daureen Nesdill, Mark Newton, Julie Speer, and Charles Watkinson. Library publishing services: strategies for success: final research report. March 2012. <http://wp.sparc.arl.org/lps/>.
- [8] Open Journal Systems. <http://pkp.sfu.ca/ojs/>.
- [9] DSpace. <http://www.dspace.org/>.
- [10] Digital Commons. <http://digitalcommons.bepress.com/>.
- [11] Sarah K. Lippincott. Library publishing directory 2014. October 2013. http://www.librarypublishing.org/sites/librarypublishing.org/files/documents/LP_C_LPDiretory2014.pdf.
- [12] HathiTrust digital library. <http://www.hathitrust.org/>.
- [13] mPach. <http://www.lib.umich.edu/mpach>.
- [14] Journal Article Tag Suite. <http://jats.nlm.nih.gov/>.
- [15] Office Open XML. Wikipedia. http://en.wikipedia.org/wiki/Office_Open_XML.
- [16] NISO Journal Article Tag Set (JATS) version 1.0: preview XSLT stylesheets. <https://github.com/NCBITools/JATSPreviewStylesheets>.
- [17] Recommended practices for online supplemental journal article materials: a recommended practice of the National Information Standards Organization and the National Federation of Advanced Information Services. January 2013. <http://www.niso.org/publications/rp/rp-15-2013>.
- [18] Collections. HathiTrust digital library. <http://babel.hathitrust.org/cgi/mb>.
- [19] HathiTrust Data API. http://www.hathitrust.org/data_api.
- [20] OpenDocument. Wikipedia. <http://en.wikipedia.org/wiki/OpenDocument>.
- [21] Book Interchange Tag Suite (BITS) version 1.0. <http://jats.nlm.nih.gov/extensions/bits/>.