ARCHIVEMATICA-EPRINTS INTEGRATION

Developing digital preservation capacity for open repositories

Tomasz Neugebauer
Concordia University
Canada
tomasz.neugebauer@concordia.ca

Sarah Lake
Concordia University
Canada
sarah.lake@concordia.ca

Abstract – Following three years of software development, requirements refinement, and testing, we released the integration of EPrints and Archivematica as a plugin to EPrints in 2021. This paper will explain how the integration evolved, how we implemented it in parallel to a new Archivematica instance at Concordia University, and how we are currently using it to preserve the contents of our institutional research repository. We will conclude with a discussion of some possible future enhancements envisioned for the integration.

Keywords – Digital preservation systems, digital repositories, integration, EPrints, Archivematica

Conference Topics – Innovation, Community

I. BACKGROUND

When Concordia University Library started planning to implement a digital preservation program in 2018, one of our first goals was to improve the digital preservation workflows for our institutional research repository (IR), Spectrum. Spectrum is built using EPrints, a free and open-source software package for creating open access repositories that are compliant with the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH).[1] Like many repository systems, EPrints' native digital preservation functionality is limited [1]. To ensure the long-term preservation of the content deposited to the IR, we needed to implement a digital preservation system that we could integrate with our EPrints repository.

After benchmarking a number of digital preservation solutions, we retained Archivematica as our preferred option. Archivematica is a project by Artefactual Systems that integrates a suite of open-source software tools allowing users to process digital objects from ingest to access in compliance with the ISO-OAIS functional model.[2] Our decision was based on Archivematica's low relative cost, its high flexibility and scalability, and its active user community that has helped develop and sustain its open-source model since 2009.

With both Archivematica and EPrints being open-source projects, we saw an opportunity to collaborate with EPrints Services, Artefactual Systems, and other members of the open-source software community to create an EPrints-Archivematica export plugin. This integration would bridge a gap between two widely-adopted open-source systems and provide valuable digital preservation functionality for EPrints repositories. It would allow us and other open repository administrators to continually ensure that files entrusted to us are not lost or corrupted and sufficient information about the digital objects is collected to enable future preservation and access.

The integration plan was first presented in 2018 at an Archivematica Camp and the Open Repositories conference [2]. Development work to

[1] https://www.eprints.org/.

[2] https://www.archivematica.org/en/.

18th International Conference on Digital Preservation
iPres 2022, Glasgow, Scotland.

create an Eprints-Archivematica export plugin started shortly thereafter, with Concordia University leading the development. We released the integration as a plugin to EPrints in 2021, and we are currently using it in production to export digital objects and metadata from Spectrum to a dedicated Archivematica instance.

This paper will explain how the current version of the plugin is being used with our Archivematica pipeline and what we learned during the development. We will conclude with a discussion of some possible future enhancements envisioned for the integration.

II. Plugin Development and Design

The GitHub repository[3] has served as the common platform for refining the specifications, and iterations of the software development. Virtual meetings were held as the plugin was developed, with participation from the community, such as the University of Strathclyde [3]. A first official release of this work as a Bazaar Plugin was completed in December 2021.

One of the fundamental questions for the export is the structure of the transfer object. The current version exports out metadata and digital objects according to the standard Archivematica transfer structure with existing checksums[4]. It uses two folders at the second level: one for metadata files and accompanying checksum file, and an objects folder with the deposited documents and derivatives.

The plugin creates and tracks transfer records for each deposit. Instead of relying on a BagIt utility for generating checksums, the plugin itself generates a checksum.md5 file that includes all of the files in the objects folder, and compares these with what is already stored inside EPrints, looking to flag any checksum missing or mismatch problems during processing.

Instead of using an eprintID and date at the top level (the initial idea from 2018), the current version of the plugin uses an "Archivematica ID" assigned by EPrints to each transfer record as the top-level folder name. This was a practical decision in that Archivematica needs to send this ID back to EPrints on completion, along with the UUID, and the callback functionality can easily retrieve it from this folder name, which is also the AIP (Archival Information Package) name inside Archivematica. In the latest release, we added the option to include a prefix for the repository name, for example "spectrum-999" instead of just the id, which is a useful configuration option for those institutions that have multiple EPrints instances exporting to the same Archivematica instance.

There are currently three command line scripts that ship with the plugin for creating, flagging, and processing. Each of these scripts can take arguments to limit its functionality to a specific deposit or a limited number of deposits. This was especially useful during development, testing, and batch processing of the existing backlog of deposits to process. After the backlog is completed, a periodic run of create and process transfers will be placed in the crontab, to export out only the newly published deposits, or ones whose files or metadata fields have changed in ways that match our specification for sufficient change to re-export. A deposit's transfer can be flagged for export by a trigger configured to watch specific metadata fields and/or changes to the uploaded files.

When Archivematica successfully processes the transfer and stores the AIP in archival storage, it sends the UUID of this transfer back to EPrints using a Service Callback. The plugin stores the UUID of the AIP in archival storage as a part of the log for each transfer record, and changes the state to "archived".

The metadata.json file generated by the plugin contains some basic descriptive metadata, including item title and date, and importantly, the eprintid/URL of the originating item. Unlike the EP3.xml file that is also included, when this is ingested by Archivematica, it is included in the METS file and indexed for searching, allowing for retrieval of AIP by eprintID, for example.

Typically, published items in repositories are either not modified post-publication, or when modified, re-published as new item versions with their own eprint iDs. However, some workflows (e.g.,

[3] https://github.com/eprintsug/EPrintsArchivematica

[4] https://www.archivematica.org/en/docs/archivematica-1.13/user-manual/transfer/transfer/#transfer-checksums

18th International Conference on Digital Preservation
iPres 2022, Glasgow, Scotland.

working papers, corrections to existing files) require that published documents change without creating a new eprint version. If the files in the item and/or metadata defined in the plugin configuration as constituting a sufficient change to merit a re-export, are modified post-export, the plugin would flag that item's archivematica record as in need of processing. It would be exported with the same top folder name again and ingested by Archivematica as a new AIP. The result would be an eprint with two separate AIPs in storage, and both the UUIDs would be associated with that item's archivematica record in EPrints.

III. Processing in Archivematica

In parallel to the plugin development, we deployed a self-hosted instance of Archivematica with an annual support contract with Artefactual Systems. Our instance currently has two pipelines, one dedicated to Spectrum, our EPrints repository, and the other to the Library's Special Collections, which also fall within the scope of the Library's digital preservation program. The following section outlines how we customized our Spectrum Archivematica pipeline to fully automate processing.

Archivematica integrates a suite of open-source tools to perform a host of preservation actions on the transferred items, including file format characterization, validation and normalization, and generating or extracting a large volume of technical preservation metadata, which is stored in a METS file. Archivematica then generates an AIP containing the transferred items and associated derivatives and metadata and places it in archival storage.

We use Automation Tools,[5] a set of Python scripts designed to automate the processing of transfers in Archivematica, to automatically check a watched folder for new transfers every three minutes. Artefactual Systems helped us implement a script to clear completed transfers from the dashboard automatically, which allowed us to process batches of several hundred transfers at a time without impacting Archivematica's performance. This modification was necessary for scalability because the accumulation of hundreds of transfers typically makes the dashboard unresponsive and prevents us from seeing the transfers in progress or troubleshooting failed transfers.

We implemented an automated processing configuration so that when a transfer is picked up by Archivematica, it is fully processed, packaged, and stored without any intervention needed on our part unless one of the microservices fails.

The only step of this workflow that isn't fully automated is deleting the completed transfers out of the transfer source folder, which we are currently doing manually through SFTP after each batch. Artefactual is currently developing a product called Enduro[6] which is intended to eventually replace Automation Tools, and could potentially help us automate this step further down the line.

For our archival storage infrastructure, Concordia University recently subscribed to the Ontario Library Research Cloud (OLRC)[7]a private cloud storage network for Canadian universities. The OLRC uses a modified version of Duracloud[8] that can be configured as a storage location in Archivematica's Storage Service application. With Duracloud, three copies of each AIP are replicated on servers with periodic fixity-checking across a private network of geographically-dispersed university-owned and operated data centers. Should one copy of an AIP become unreadable, it is automatically replaced by a new one created from the two others.

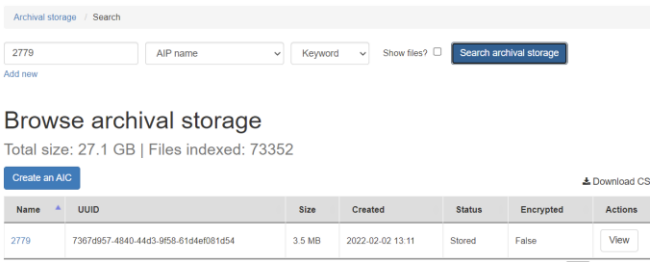

Figure 1 EPrintsArchivematica export in Archival Storage through the Archivematica Dashboard. Showing results of search by "AIP Name", which is also the Archivematica object ID for this item inside EPrints.

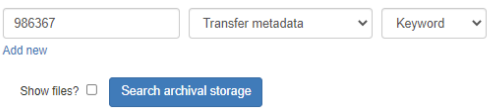

Figure 2 It is also possible to search by descriptive metadata submitted in the metadata.json file in the transfer, such as the EPrintID. For this, the "Transfer metadata" index is used on the "Browse Archival storage" interface in Archivematica Dashboard.

[5] https://github.com/artefactual/automation-tools.

[6] https://github.com/artefactual-labs/enduro.

[7] https://cloud.scholarsportal.info/.

[8] https://www.lyrasis.org/DCSP/Pages/DuraCloud.aspx.

18th International Conference on Digital Preservation
iPres 2022, Glasgow, Scotland.

Once an eprint AIP has been created and stored, a post-store callback in Archivematica updates the item record in EPrints to include the Archivematica UUID of the eprint and a timestamp. This serves as a confirmation in both systems that the item has been preserved–a feature that we originally thought would be nice to have and that we ultimately decided was essential. A caveat to this is that the callback cannot be limited to the pipeline that needs it, which isn't ideal for institutions using one storage service for two pipelines like we are. We have proposed this as a feature on the Archivematica GitHub.[9] Not being able to limit the callback to a specific pipeline means that on the EPrints side, we have to ignore irrelevant callbacks from the storage service when it places Special Collections AIPs in archival storage. It is relatively simple to enable or disable a specific callback using the Storage Service, so we can disable it during times when the other (Special Collections) pipeline is actively processing.

AMID	Dataset ID	DataObj ID	Needs update	UUID
2779	eprint	986367	No	7367d957-4840-44d3-9f58-61d4ef081d54

Timestamp	Action	Comment	Result
23 January 2022 01:00:33 UTC	Transfer Created	created via trigger	Success
2 February 2022 18:06:48 UTC	Transfer Processed	processed	Success
2 February 2022 18:11:14 UTC	Transfer Archived	UUID:[7367d957-4840-44d3-9f58-61d4ef081d54]	Success

Figure 3 Archivematica Records Management Screen in EPrints viewing the details of a transfer. The UUID of the AIP in Archivematica is sent to EPrints using a Callback.

IV. Issues and Limitations

OpenDOAR[10] reports more than 630 instances of EPrints repositories worldwide, some of which might very well be running an out-of-date EPrints platform and in need of digital preservation. More testing of the plugin on out-of-date EPrints repository versions might prove valuable in the future. The plugin should work well for EPrints instances running any version of Eprints 3.3 or 3.4. It was developed and tested successfully on EPrints version 3.3.12 and 3.4.3. The authors are not aware of any tests of the plugin on EPrints repositories running earlier versions than 3.3.12. On the Archivematica side, the integration was developed and tested on Archivematica 1.12 / Storage Controller 0.16, but we have since upgraded to Archivematica 1.13 / Storage Controller 0.18.

As we started to use the plugin in production, we came across unanticipated issues and limitations that prompted us to reflect on our preservation strategy as a whole and led us to make changes to the code and our workflow. We have been using this plugin to preserve more than a decade's worth of deposits from our institutional repository, and one of the discoveries we made in this process is that thousands of PDFs imported into the repository did not have a checksum in the EPrints database. The issue was so common that we decided to develop functionality in the plugin to add the checksums to EPrints during export, throwing only a warning rather than the usual "checksum mismatch" error in these cases.

Another lesson-learned about EPrints during the development of the plugin is that the file names stored in the the EPrints File Object[11] can in some cases differ from the corresponding file names on disk. The use of regular expressions was required in the plugin to replace some characters (quote and double-quote) in the value that is returned by the internal EPrints object call to get "filename". The file's URL is served over HTTPS with the quotes and double-quotes in the name, but stored on disk with a file name that replaces those characters with their ASCII addresses: =0027 and =0022.

We also found that any metadata-only eprints, i.e. items that did not contain a document or upload of any kind, caused the transfer to fail in Archivematica. This issue prompted a discussion about whether or not these items should be preserved in Archivematica at all, since they didn't contain any deposited digital objects. In the end, we added an option in the plugin to export the metadata-only transfers to a different folder location so that they could get skipped over by Automation Tools. The metadata would still get exported out to be stored along with any of the logs from the preservation batch jobs.

One limitation of our workflow is that we have found the process of resolving certain errors to be unnecessarily labor-intensive. For instance, if a normalization job fails due to an error with the tool, command, or file, our automatic processing configuration will approve the normalization anyway. In these cases, we have had to re-ingest the transfer with a manual normalization workflow, which involves adding the preservation derivative to the transfer folder through SFTP, generating a

[9] https://github.com/archivematica/Issues/issues/1325.

[10] https://v2.sherpa.ac.uk/opendoar/.

[11] https://wiki.eprints.org/w/File_Object

18th International Conference on Digital Preservation
iPres 2022, Glasgow, Scotland.

checksum, and adding it to the checksum file; otherwise, the transfer will fail due to mismatched checksums. We have only encountered this issue in a couple of transfers so far so it has not been a significant issue, but it would be helpful to have a more efficient way of handling this.

Most of the preservation issues we encountered as we processed our repository's backlog involved transfers containing unusual deposit formats. For example, in instances where a student had created software as their thesis project, the application bundle in their deposit often contained files that would cause the transfer to fail in Archivematica. In one case Archivematica was unable to assign UUIDs to symbolic links in a MacOS application bundle, which made the transfer fail at the "Generate METS" step. Our solution was to simply not extract the problematic packages, which isn't perfect, but it allows us to perform bit-level preservation as a first pass and leaves us the possibility to revisit content-level preservation in the future.

V. Conclusion

In spite of the occasional issues and limitations, we are very satisfied with the results of this integration project. As of the publication of this paper, we have successfully processed the entire backlog of roughly 18,000 live eprints from our repository, resulting in approximately 500,000 indexed files in archival storage. We are in the process of determining what our preservation workflow will be going forward. This will most likely involve running create and process transfers weekly and establishing a procedure for the removal of items from archival storage. This will be done in coordination with the Thesis Office who would communicate with the Digital Preservation Librarian if, for example, a thesis is withdrawn.

A future enhancement to the plugin is to include in the AIP the processing log generated from the results of each command on each eprint that was exported using the plugin.[12] We determined that since this log is a record of the preservation actions that were performed upon the objects, it should be encoded as preservation metadata using PREMIS. PREMIS is a de facto digital preservation metadata standard implemented in Archivematica and supported to some degree in many other systems such as: Islandora, RODA, Preservica, DSpace, BitCurator, AtoM, HathiTrust [4]. Archivematica can parse an imported premis.xml file into the main METS file of the AIP, so that all of the preservation metadata about the objects is in one place and in a machine-readable and interoperable format.

We anticipate that feedback from the user community will inspire new enhancements that will allow this integration to flourish over time. For example, the community has already identified the need for a delete_transfers script that would make it easier to remove transfer objects from EPrints that are no longer needed. We see the release of this plugin as an important step towards bridging gaps in open-source digital preservation workflows for repositories, and we hope that it will empower repository managers to implement digital preservation practices at their institutions.

ACKNOWLEDGMENT

The authors would like to thank EPrints Services, for their development work and collaboration that made this project possible, and Concordia University Library for sponsoring the work. We would also like to thank the members of the EPrints open-source community for their contributions to and feedback on the plugin, and Artefactual Systems for the support and guidance they provided for the configuration of our Archivematica pipeline and Automation Tools. We would also like to thank Tessa Walsh for her thoughtful suggestions and contributions in the development of this plugin and Concordia University's digital preservation planning.

REFERENCES

[1] T. Neugebauer, P. Lasou, A. Kosavic, and T. Walsh, "Digital Preservation Functionality in Canadian Repositories," Canadian Association of Research Libraries. 2019. Available:https://www.carl-abrc.ca/wp-content/uploads/2019/12/orwg_report2_preservation_repos_en.pdf.

[2] T. Neugebauer, J. Bradley, and J. Simpson, "Digital Preservation through EPrints-Archivematica Integration," *International Conference on Open Repositories*, Bozeman, USA, 2018. Available: https://spectrum.library.concordia.ca/id/eprint/983933/.

[3] G. Macgregor and T. Neugebauer, "Preserving digital content through improved EPrints repository integration with Archivematica," *UK Archivematica User Group*, 2020. Available: https://strathprints.strath.ac.uk/73978/.

[12] https://github.com/eprintsug/EPrintsArchivematica/issues/37.

[4] M. Jordan and E. McLellan, “PREMIS in Open-Source Software: Islandora and Archivematica,” in *Digital Preservation Metadata for Practitioners*. Cham., Switzerland: Springer, 2016, ch. 16, pp. 227-239. [Online]. Available: https://doi.org/10.1007/978-3-319-43763-7_16.

18th International Conference on Digital Preservation
iPres 2022, Glasgow, Scotland.