Harmonizing Heterogeneous WARC Contributions: Normalization Lessons from the 2024 End of Term Web Archive

Sawood Alam
Internet Archive
San Francisco, CA, USA
sawood@archive.org

Mark E. Phillips
University of North Texas
Denton, TX, USA
mark.phillips@unt.edu

Abstract

The End of Term Web Archive (EOT) documents the United States federal web presence during presidential transitions. The 2024 transition produced the largest EOT collection to date: more than 2.3 petabytes captured by six partners using a variety of tools and workflows. Contributions included record-at-a-time gzip-compressed WARCs, WARCs packaged within WACZ, Zstandard-compressed WARCs, and WARCs embedded in BagIt structures that separate headers from payloads. While each approach was valid locally, their coexistence complicated preservation management, derivative generation, and large-scale computational use. To provide a consistent, researcher-friendly corpus, the EOT team normalized these contributions into individual record-at-a-time gzip-compressed WARC files by unpacking WACZ packages, recompressing Zstandard WARCs, reconstructing BagIt-based captures, and salvaging usable records from partially corrupted files. We describe the workflows, tools, and validation procedures used and the organization of the resulting dataset, and reflect on the tradeoffs between workflow diversity and downstream interoperability in collaborative web archiving.

1 Introduction

Every four years, the End of Term Web Archive (EOT) captures a snapshot of the United States federal web during the transition of executive power. By harvesting federal `.gov` and public `.mil` domains, the project records how policy, priorities, and public communication change as one administration hands over to the next. The most recent transition, from the Biden to the Trump administration, produced the largest EOT collection to date: more than 2.3 petabytes of content crawled by six partner institutions.

From the start, EOT has been a distributed, collaborative effort [22]. Each partner uses its own tools, workflows, and curatorial strategies and focuses on content that fits its mission and technical capacity. This diversity has long been seen as a strength of the project, and it has made room for experimentation in large-scale web archiving. At the scale of the 2024 collection, however, differences in how partners created and packaged Web ARChive (WARC) files [12] caused real problems for preservation and access.

Partners contributed WARC files compressed record-at-a-time with gzip, WARC files packaged in the Web Archive Collection Zipped (WACZ) format, WARC files compressed with Zstandard, and WARC files split across BagIt packages, with headers and payloads stored separately. Each of these choices made sense locally, but together they made downstream computational use and large-scale analysis harder.

To give researchers a consistent corpus, the EOT team converted all contributions into individual WARC files with record-at-a-time gzip compression. This meant working through technical and structural differences between formats without compromising the original data. The normalized dataset is publicly available on AWS S3 [3], and the originally deposited formats remain preserved at the Internet Archive [14], where they are accessible through the Wayback Machine.

This paper is an abridged version of our iPRES 2026 paper [21]. We describe the technical decisions, tradeoffs, and workflows behind the normalization and share lessons for other large, collaborative web archiving projects.

2 Background

EOT crawling is coordinated through the URL Nomination Tool, hosted by the University of North Texas (UNT) Libraries, which collects nominated URLs and bulk seed lists from partners and the public. The 2024 cycle combined traditional large-scale crawlers (Heritrix, GNU Wget, Apache Nutch) with browser-based capture tools such as Browsertrix, pairing the depth of long-running frontier crawls with the replay quality of browser-based capture. Crawling began in mid-September 2024 with an Internet Archive "book-end" crawl [22] and biweekly UNT crawls of human-nominated seeds, shifted to targeted documentation of the transition after the election, and ended in late April 2025, a cutoff consistent with earlier EOT collections (2008, 2012, 2016, and 2020) [19]. Funding from Bloomberg Philanthropies to EOT partner Environmental Data and Governance Initiative paid for additional crawling, including captures of complex, dataset-rich interfaces. The Harvard Law School Library Innovation Lab (LIL), working with the Data Rescue Project [7], harvested datasets from federal aggregators such as `data.gov`.

After crawling ended, the contributions were consolidated into two complete copies: one at the Internet Archive (IA) [14], where the content is also replayed in the Wayback Machine, and one in Amazon Web Services (AWS) S3 [3], organized for large-scale computational access. Data reached the two copies by different routes (Table 1). Data already at IA, or sent there first, was copied to S3; this covers the contributions of IA, Archive Team, and UNT. LIL had published its data in its own AWS bucket, from which we copied it to IA, converted it to WARC there, and moved the resulting files to the EOT S3 bucket. Common Crawl and Webrecorder uploaded directly to the EOT bucket, and their data was then copied to IA. IA's Petabox storage keeps files in items with limits on file count and total size, so the WARC files from a single Common Crawl S3 segment had to be rebucketed into several IA items. Each normalization step ran wherever it was most practical: Zstandard recompression at UNT, and WACZ unpacking and BagIt conversion at IA. Afterwards, the Zstandard and WACZ files that had been uploaded to S3 were removed, so S3 holds only `warc.gz` files while

Table 1: The End of Term 2024 WARC dataset on AWS S3, with contributed formats and their movements to ensure copies at the Internet Archive (IA) and AWS S3; braces mark normalization steps.

Crawl Partner	Files	Size	Contributed Format	Data Movement
Archive Team	136,697	1.4 PB	WARC.gz	IA → S3
			WARC.zst	IA → S3 → UNT {`zst-to-gz`} → S3 (WARC.zst removed from S3)
Common Crawl Foundation	1,529	1.5 TB	Repackaged WARC.gz	S3 → IA (rebucketed)
Harvard Library Innovation Lab	311,604	16 TB	BagIt.zip	LIL S3 → IA {`bagit-to-warc`} → S3
Internet Archive	699,060	853 TB	WARC.gz	S3
UNT Libraries	41,700	46 TB	WARC.gz	IA → S3
			WACZ	IA {`wacz-to-warc`} → S3
Webrecorder	26,301	11 TB	WACZ	S3 → IA {`wacz-to-warc`} → S3 (WACZ removed from S3)
Total	1,216,891	2.3 PB		

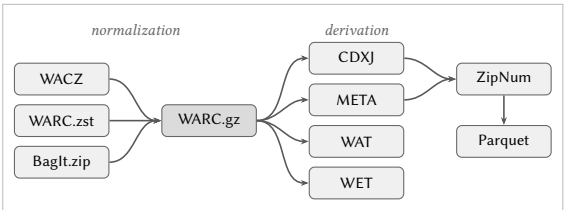


Figure 1: Normalization and derivation transitions. Every source format not already in the baseline `warc.gz` format was converted to it, and all derivatives were then produced from the unified `warc.gz` files.

IA keeps the formats as deposited. From the S3 copy, we generated WAT, WET, CDXJ, and WARC Metadata Sidecar (META) [18] files for each WARC file, merged the sidecar metadata into the CDXJ records, and built a ZipNum index for replay, from which a Parquet dataset for SQL-based analysis was derived [20]. Since every derivative is produced per WARC file, the WARC file is the basic unit of the dataset (Figure 1).

3 Related Work

Deriving smaller files from WARC data is well-established practice in web archiving. Besides CDX indexes for lookup and replay, WAT and WET files are commonly generated for each WARC file and named after it, so users can tell which WARC file each derivative came from [18]. Derivatives are also much smaller than their source. In the EOT-2020 crawl, for example, a 953.7 MiB WARC file has WAT, WET, META, and CDX files of 449.5, 82.7, 40.5, and 3.9 MiB, respectively [18]. The Archives Unleashed Project has built tools and services around such derivatives [5], and national programs such as the Library of Congress and the UK Web Archive publish sample datasets and derivatives, which also allow non-consumptive use of collections that cannot be shared directly for rights reasons [17, 24]. Common Crawl is the closest model for the EOT dataset. It publishes monthly crawls as WARC files together with WAT, WET, CDX, and Parquet indexes, and records language, character set, and MIME type at crawl time [6]. Summaries of CDX indexes [2] offer another compact way to describe what a collection contains. All of these workflows assume consistent, well-formed WARC input, which the EOT 2024 contributions did not initially provide.

4 Partner Contributions

Table 1 summarizes the corpus. Which normalization pathway a contribution needed depended largely on the partner and its crawling methods.

Internet Archive. The Internet Archive contributed four major crawls. Three were large-scale Heritrix crawls split by period: about 124 TB collected before the election, 251 TB between the election and the inauguration, and 261 TB after the inauguration. The fourth was 218 TB of video from official U.S. federal YouTube channels.

University of North Texas Libraries. UNT contributed 46 TB from three kinds of crawls: biweekly Heritrix and Browsertrix crawls of human-nominated seeds, broad but shallow Browsertrix crawls of more than 300 federal agency domains to improve replay fidelity, and long-running Heritrix crawls of data-intensive sites that benefited from sustained frontier expansion.

Common Crawl Foundation. Common Crawl extracted `.gov` and `.mil` records from its public web crawls between September 2024 and April 2025 and repackaged them into new WARC files. The extract consists mostly of HTML and PDF resources, without embedded assets such as stylesheets, scripts, images, or fonts.

Harvard Law School Library Innovation Lab. LIL contributed over 16 TB from its effort to download and preserve more than 300,000 public federal datasets considered at risk during the transition. The data, packaged in BagIt format [16], covers structured datasets that conventional crawling can easily miss.

Webrecorder. Webrecorder used Browsertrix to capture dynamic, data-rich websites that traditional crawlers handle poorly, producing 10.5 TB of content packaged as WACZ.

Archive Team. Archive Team contributed several large collections from its wget-based crawling infrastructure: about 823 TB of general `.gov` and `.mil` crawl data, 217 TB from the Defense Visual Information Distribution Service (DVIDS), and 393 TB from Voice of America properties.

5 Methodology

Three objectives guided the normalization. The first was technical standardization. As in earlier EOT collections, record-compressed WARC (`warc.gz`) files are the primary access unit, with predictable directory structures and naming conventions and strict adherence to specifications. This limits format drift and makes automated validation, derivative generation, and computational analysis possible.

The second was the experience of users and researchers. Consistent handling of identifiers, record boundaries, and encodings across institutions lowers the barrier for people unfamiliar with idiosyncratic implementations, and we deliberately avoided adding new tools to the access chain. The third was long-term sustainability. Keeping a clear line between authoritative preservation objects and derived assets, and documenting the process, reduces technical debt and makes future EOT cycles easier to repeat.

5.1 Normalization Pathways

We used three normalization pathways (Figure 1, left): unpacking WACZ files, recompressing Zstandard WARC files with gzip, and rebuilding BagIt-based captures as standard WARC files.

5.1.1 WACZ Unpacking. WACZ [15] is a packaging format, specified by Webrecorder, that is increasingly well supported for distributing web archive collections. A single WACZ package can bundle several WARC files, along with CDXJ indexes and metadata, in an uncompressed ZIP container, which allows efficient access to the contents and playback directly from the package.

This model works well for distribution and playback. Our derivatives, however, are generated per WARC file, and treating the WACZ package as the unit of the dataset would have complicated both processing and the organization of the outputs.

We therefore unzipped each WACZ package, identified the WARC files in its archive directory, and ingested them individually into AWS storage, where downstream processing and derivative generation could work on each file directly. The original WACZ package was kept in the corresponding Internet Archive collection to preserve the original distribution format.

5.1.2 Zstandard Recompression. Zstandard is a fast lossless compression algorithm introduced at Facebook in 2015 and designed in particular for faster decompression than gzip. Because it can reduce storage requirements for some web archive content, it has attracted interest in the web archiving community, and a draft specification describes its use in WARC files [13]. It is not yet widely used, but Archive Team is actively exploring it in its crawling.

Zstandard-compressed WARC files need more processing than the gzip-compressed files common in web archiving workflows. In a gzip-compressed WARC file, any record can be read directly using its offset and compressed length. A Zstandard-compressed WARC file, in contrast, carries a compression dictionary in its header, and that dictionary must be retrieved before any record can be decoded.

Our workflow downloads each Zstandard-compressed WARC file from AWS, extracts the dictionary from its header with a Python script, and decompresses the file with the `zstd` utility. The uncompressed WARC file is then recompressed record by record with gzip using the FastWARC toolkit, producing the format that downstream access and indexing workflows expect.

Each recompressed file is then validated and indexed. Generating its WARC Metadata Sidecar reads every record with warcio, which confirms that the file parses and that recompression introduced no structural errors. The `cdxj-indexer` tool then builds a CDXJ index for the file. Only if every step succeeds is the `warc.gz` file uploaded back to AWS.

Of the 80,762 Zstandard-compressed WARC files processed this way, 80,715 were recompressed, validated, and indexed successfully; only 47 failed.

5.1.3 BagIt WARC Generation. The LIL data [9] was created with the bag-nabit tool [10], which stores the captured request and response headers in a WARC file but keeps some payloads as ordinary files inside the BagIt package. We wrote `bagnabit2warc` [1] to rebuild one standard WARC file from each of the more than 300,000 packages while preserving as much of the original capture metadata as possible. Each input is a bag-nabit ZIP package with a `data/headers.warc` file, which holds the original request, response, revisit, and other records, and payload files under `data/files/*` that belong to certain revisit records.

The output WARC file starts with a `warcinfo` record that describes the source package and the conversion. If the package includes `data/signed-metadata.json`, a `metadata` record derived from it follows. All records from `data/headers.warc` are carried over with their WARC and HTTP headers intact. For revisit records whose payloads are stored in `data/files/*`, the tool reconstructs the corresponding response records by combining the stored payload with the original revisit metadata. Apart from the two added records, existing WARC headers, record identifiers, and timestamps are left unchanged.

Payload and block digests are recomputed for every record in the new file. On reconstructed records, HTTP headers such as Content-Encoding, Transfer-Encoding, and Content-Length do not always match the rehydrated payload, because the stored payload files may reflect earlier processing, such as partial decoding of content encodings. We keep the original HTTP headers anyway, so that the records reflect what was actually captured, rather than rewriting them to fit the payload. Each conversion is logged in JSON Lines format, with counts of the records processed by type and any transformations applied, so that conversions can be audited and reproduced.

5.2 Recovering WARC Data

A recurring problem during dataset preparation was WARC files that could not be processed because of structural or compression errors. Many web archiving tools read a partially corrupted WARC file until they reach an unrecoverable error and then stop. Because a WARC file is a sequence of concatenated records, at least part of such a file usually remains usable for playback or analysis. In large collections, this kind of partial loss is often tolerated because a few inaccessible records have little overall impact. For EOT, however, derivatives are expected for every WARC file, so a file that failed to process cleanly often left missing or incomplete derivatives behind.

We analyzed the errors in the problematic files and developed procedures to recover as much valid data as possible. Many errors were compression failures in individual records: when gzip or Zstandard could not decompress one record, the whole pipeline failed. We wrote a pair of scripts [4] that walk through a WARC file and try to decompress each record on its own. Records that decompress are written to a recovered WARC file. When a record fails, the script skips it and scans ahead to the next valid record boundary. In this way we recover valid records that conventional tools lose by stopping at the first error.

```
crawl-data/EOT-2024/segments
                    |-- AT-{000..014}
                    |   |-- cdx
                    |   |-- meta
                    |   |-- warc
                    |   |-- wat
                    |   `- wet
                    |-- CC-000
                    |-- HIL-{000..031}
                    |-- IA-{000..071}
                    |-- UNT-{000..004}
                    `- WR-{000..002}
```

Figure 2: EOT-2024 directory structure with segment folders (blue) and file type folders (red); the file type folders are shown only once for brevity.

Recovery started by downloading the affected file from AWS and comparing its MD5 checksum with that of the copy at the Internet Archive. If the checksums matched, recovery proceeded on the AWS copy. If they differed, the Internet Archive copy was tested and, if intact, replaced the AWS copy. For `warc.gz` files, we first tried to decompress and validate the file with Python's gzip library, warcio, and FastWARC. If these tools failed, the recovery scripts extracted the individual records that formed valid gzip members, and the result was checked again with warcio and FastWARC to make sure it parsed correctly.

Transfers between systems were verified by comparing file sizes and checksums. Recovered files are slightly smaller than the originals because damaged records are dropped, so their checksums no longer match the source files. To keep them clearly distinguishable, recovered files were renamed to mark them as repaired. Together, these steps keep broken records and degraded files away from downstream users.

6 Dataset Organization

The normalized dataset is stored in the `eotarchive` S3 bucket under `crawl-data/EOT-2024/segments` [3] (Figure 2). Each partner's contribution is split into numbered segments of manageable size, named with a short partner prefix (for example, `AT-000` to `AT-014` for Archive Team, and `HIL` for LIL). Each segment holds one folder for the primary WARC files and one for each of the four per-WARC derivatives. Derivative files keep the name of their WARC file and change only the extension, from `.warc.gz` to `.cdxj.gz`, `.warc.meta.gz`, `.warc.wat.gz`, or `.warc.wet.gz`. Users can download as much or as little as they need, from the whole collection down to one segment or one file type, over plain HTTP or with any S3 client. The file formats are the same as in earlier EOT datasets [18].

WARC. Most of the data is in WARC, the ISO standard format (ISO 28500) for storing harvested web content [12]. A WARC file is a series of records of types such as `warcinfo`, `request`, `response`, `resource`, `revisit`, and `metadata`, each compressed separately with gzip and concatenated, so tools can read any single record without decompressing the rest of the file.

WAT. Web Archive Transformation (WAT) files hold metadata and link structures extracted from the corresponding WARC file, including outgoing links and their anchor text. They suit tasks such as building link graphs, where the full content of a resource is not needed.

WET. Web Extracted Text (WET) files hold the plain text extracted from HTML and text resources. They contain no records for other content types.

CDXJ. CDXJ files index the records of a WARC file, one line per record. Each line starts with a Sort-friendly URI Reordering Transform (SURT) key and a timestamp, followed by a JSON object with fields such as the original URL, HTTP status code, MIME type, payload digest, record length, and offset in the WARC file. Sorted and merged, these indexes drive replay systems such as OpenWayback [11] and pywb [26].

META. WARC Metadata Sidecar (META) files [25] store content-based characterizations of `response` and `resource` records, including character set, language, file format, and MIME type identification and soft-404 detection. The results are kept in WARC `metadata` records. WAT and WET files distill data that is already in the records, whereas META files add information produced by characterization tools.

Collection-level indexes. The ZipNum index splits the merged, META-enriched CDXJ records into compressed blocks with a small summary file for lookups, which supports replay directly from S3. The Parquet dataset [23] derived from it supports SQL queries over the whole collection [20]. It answers questions that would otherwise require scanning the WARC files, such as which MIME types servers most often misreport, how much non-English content the archive holds and in which languages, and which domains return the most soft-404 pages [18].

7 Lessons Learned and Conclusion

Normalizing the data from six EOT partners taught us several lessons. First, expectations for web archive outputs are best agreed on before the data is created. Fixing implementation differences once they are embedded in WARC files takes substantial time and effort, and the cost grows with each EOT collection. EOT has always let partners use the tools and workflows that suit their local goals and environments, and that flexibility remains central to the collaboration. Our experience shows, however, that a common output format would greatly reduce the work needed to prepare the data for downstream users. Second, several contributed formats required specialized tools and knowledge beyond the usual access patterns for gzip-compressed WARC files. Normalizing them removes that barrier for researchers and practitioners who want to use the dataset. Finally, the tools developed in this work, including `bagnabit2warc` and the `warc-recovery` scripts, may help others normalize, repair, and process web archive data at scale.

Acknowledgments

The End of Term Web Archive is sustained through the contributions of time, expertise, and resources from its partner organizations [8]. The Filecoin Foundation for the Decentralized Web (FFDW) and Bloomberg Philanthropies provided financial support for this work. Staging the dataset was also supported by the Institute of Museum and Library Services grant LG-252349-OLS-22. We gratefully acknowledge hosting and infrastructure support from the AWS Open Data Program.

References

[1] Sawood Alam. 2026. bagnabit2warc. https://github.com/internetarchive/bagnabit2warc.

[2] Sawood Alam and Mark Graham. 2022. CDX Summary: Web Archival Collection Insights. In *Linking Theory and Practice of Digital Libraries - 26th International Conference on Theory and Practice of Digital Libraries, TPDL 2022, Padua, Italy, September 20-23, 2022, Proceedings (Lecture Notes in Computer Science, Vol. 13541)*, Gianmaria Silvello, Óscar Corcho, Paolo Manghi, Giorgio Maria Di Nunzio, Koraljka Golub, Nicola Ferro, and Antonella Poggi (Eds.). Springer, 297–305. doi:10.1007/978-3-031-16802-4_25

[3] Sawood Alam and Mark Edward Phillips. 2026. End of Term 2024 Dataset. https://eotarchive.org/data/data-2024/

[4] Sawood Alam and Mark E. Phillips. 2026. warc-recovery. https://github.com/end-of-term/warc-recovery.

[5] Archives Unleashed. 2022. The Archives Unleashed Project. https://archivesunleashed.org/.

[6] Common Crawl. 2018. August Crawl Archive Introduces Language Annotations. https://commoncrawl.org/2018/08/august-2018-crawl-archive-now-available/.

[7] Data Rescue Project. 2025. Data Rescue Project. https://www.datarescueproject.org/.

[8] End of Term Web Archive. 2025. End of Term Web Archive. https://eotarchive.org/.

[9] Harvard Library Innovation Lab. 2024. Archive of Data.gov. https://source.coop/harvard-lil/gov-data.

[10] Harvard Library Innovation Lab. 2024. bag-nabit. https://github.com/harvard-lil/bag-nabit.

[11] International Internet Preservation Consortium. 2021. OpenWayback. https://github.com/iipc/openwayback.

[12] International Internet Preservation Consortium. 2022. The WARC Format 1.0. https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.0/.

[13] International Internet Preservation Consortium. 2023. Zstandard Compression for WARC Files 1.0. https://iipc.github.io/warc-specifications/specifications/warc-zstd/.

[14] Internet Archive. 2024. End of Term 2024 Web Crawls. https://archive.org/details/EndOfTerm2024WebCrawls.

[15] Ilya Kreymer and Ed Summers. 2021. Web Archive Collection Zipped (WACZ). https://specs.webrecorder.net/wacz/1.1.1/.

[16] John A. Kunze, Justin Littman, Liz Madden, John Scancella, and Chris Adams. 2018. The BagIt File Packaging Format (V1.0). https://www.rfc-editor.org/info/rfc8493. doi:10.17487/RFC8493

[17] Library of Congress. 2023. Web Archive Datasets. https://labs.loc.gov/work/experiments/webarchive-datasets/

[18] Mark E. Phillips, Kristy K. Phillips, and Sawood Alam. 2023. *Content-based Characterization of the End of Term Web Archive.* International Conference on Digital Preservation, 1–7. https://hdl.handle.net/2142/121091

[19] Mark E. Phillips, Kristy K. Phillips, and Sawood Alam. 2024. End of Term Web Archive Dataset: Longitudinal Web Archive of .GOV and .MIL Domains. In *Proceedings of the 2023 ACM/IEEE Joint Conference on Digital Libraries* (Santa Fe, New Mexico, USA) *(JCDL '23)*. IEEE Press, 98–101. doi:10.1109/JCDL57899.2023.00024

[20] Mark E. Phillips, Kristy K. Phillips, and Sawood Alam. 2025. End of Term Web Archive Dataset: Documenting the 2024 Presidential Transition. In *2025 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. 293–296. doi:10.1109/JCDL67857.2025.00049

[21] Mark E. Phillips, Kristy K. Phillips, and Sawood Alam. 2026. Lessons Learned in Normalization for Access in the 2024 End of Term Web Archive. In *Proceedings of the 22nd International Conference on Digital Preservation (iPRES 2026)*.

[22] T. Seneca, A. Grotke, C. N. Harman, and K. Carpenter. 2012. It Takes A Village To Save The Web: The End Of Term Web Archive. *Documents to the People* 40, 1 (2012), 1–21. https://digital.library.unt.edu/ark:/67531/metadc84373/

[23] The Apache Software Foundation. 2023. Apache Parquet. https://parquet.apache.org/

[24] UK Web Archive. 2023. UK Web Archive Open Data. https://data.webarchive.org.uk/opendata/

[25] University of North Texas Libraries. 2023. WARC Metadata Sidecar. https://github.com/unt-libraries/warc-metadata-sidecar.

[26] Webrecorder. 2023. pywb. https://github.com/webrecorder/pywb.