ArchiveNow: Simplified, Extensible, Multi-Archive Preservation

Mohamed Aturban, Mat Kelly, Sawood Alam, John A. Berlin,
Michael L. Nelson, and Michele C. Weigle
Old Dominion University
Department of Computer Science
Norfolk, Virginia, USA
{maturban,mkelly,salam,jberlin,mln,mweigle}@cs.odu.edu

ABSTRACT

ArchiveNow is a Python module for preserving web pages in on-demand web archives. This module allows a user to submit a URI of a web page for archiving at several configured web archives. Once the web page is captured, *ArchiveNow* provides the user with links to the archived copies of the web page. *ArchiveNow* is initially configured to use four archives but is easily configurable to add or remove other archives. In addition to pushing web pages to public archives, *ArchiveNow*, through the use of *Wget* and *Squidwarc*, allows users to generate local WARC files, enabling them to create their own personal and private archives.

CCS CONCEPTS

• **Information systems → Digital libraries and archives**; **World Wide Web**;

KEYWORDS

Web Archiving, Memento, WARC

ACM Reference Format:
Mohamed Aturban, Mat Kelly, Sawood Alam, John A. Berlin, Michael L. Nelson, and Michele C. Weigle. 2018. ArchiveNow: Simplified, Extensible, Multi-Archive Preservation. In *JCDL '18: The 18th ACM/IEEE Joint Conference on Digital Libraries, June 3–7, 2018, Fort Worth, TX, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3197026.3203880

1 INTRODUCTION

Preserving a web page in only a single web archive is risky. Archives may be vulnerable to security threats, or may also become temporarily or permanently unreachable. Thus, preserving web pages in multiple web archives should decrease the danger of losing archived data. Some existing work focuses on preserving web pages in multiple archives. Kelly et al. [6] built *Mink*, a Google Chrome extension that notifies a user of any available archived copies for a viewed page and suggests to archive the page in three archives. Welsh [10] developed several tools intended to archive news-related resources. For example, Welsh's *Savemy.news* (`www.savemy.news`) saves web pages in two archives. Users of this service are required to create accounts. In addition to *Savemy.news*, Welsh built three separate Python libraries to interact with on-demand archiving services. *Webrecorder* [9] and *WARCreate* [7] can be used to generate WARC files [5], but the only way to use these tools is through a web browser. *ArchiveNow* can save pages in four web archives, generate WARC files, and allows customization of the set of archives used to preserve the web. ArchiveNow does not require users to have an account and can be run through the command-line (CLI), a web-based user interface (UI), a self-contained Docker container, or as a Python module. *ArchiveNow* is available for download at https://github.com/oduwsdl/archivenow.

JCDL '18, June 3–7, 2018, Fort Worth, TX, USA

ACM ISBN 978-1-4503-5178-2/18/06.
https://doi.org/10.1145/3197026.3203880

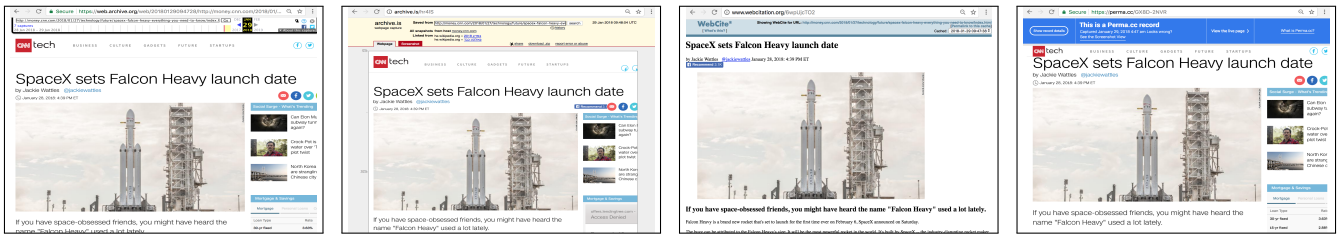
```
% archivenow --all --cc_api_key=7e..3f http://mon
ey.cnn.com/2018/01/27/technology/future/spacex-fa
lcon-heavy-everything-you-need-to-know/index.html

{
  "uri": "http://money.cnn.com/2018/01/27/technol
  ogy/future/spacex-falcon-heavy-everything-you-n
  eed-to-know/index.html",
  "request-datetime": "20180129094723",
  "mementos": {
    "archive.org": "https://web.archive.org/web/2
    0180129094728/http://money.cnn.com/2018/01/27
    /technology/future/spacex-falcon-heavy-everyt
    hing-you-need-to-know/index.html",
    "archive.is": "https://archive.is/hr4lS",
    "perma.cc": "https://perma.cc/GX8D-2NVR",
    "webcitation.org": "http://www.webcitation.or
    g/6wpUjcTO2"
  }
}
```

Figure 1: CLI example for archiving a web page in all configured web archives

2 MULTI-ARCHIVE PRESERVATION

ArchiveNow by default is configured to accept a URI from a user for archiving at the following four archives: the Internet Archive (IA) at `archive.org`, Archive.is (`archive.is`), Perma (`perma.cc`), and WebCite (`webcitation.org`). Figure 1 shows an example of running *ArchiveNow* to request capturing a web page by all configured archives. The value of `--cc_api_key` is an API key required by Perma. The user can select one or more archives by replacing `--all` with the corresponding archives' identifiers, such as `--ia` for IA, `--is` for Archive.is, `--wc` for WebCite, and `--cc` for Perma. Figure 2 shows the archived pages (mementos) from the four archives. In addition to running *ArchiveNow* via CLI, it can be run as a Docker container, a web service, or in Python. For example, after running "`archivenow --server`", a user can open `http://0.0.0.0:12345` in a web browser and use the UI page shown in Figure 3. A full list of options for running *ArchiveNow* is available on GitHub [1].



(a) In the Internet Archive. (b) In Archive.is. (c) In WebCite. (d) In Perma.

Figure 2: The web page http://money.cnn.com/2018/01/27/technology/future/spacex-falcon-heavy-everything-you-need-to-know/index.html as it appeared on the web on January 26, 2018 is archived by four different archives using *ArchiveNow*.

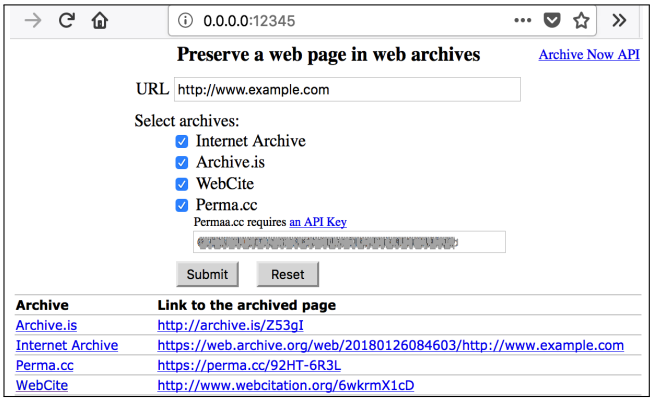


Figure 3: The UI page generated by *ArchiveNow*.

3 GENERATING WARCS

ArchiveNow can also generate a WARC file using two different tools, *Wget* [3] and *Squidwarc* [2]. The WARC file format is commonly used in the web archiving community. Archived web pages stored in WARC files can be played back by tools such as OpenWayback [4] or PyWb [8]. The feature of creating WARC files by *ArchiveNow* is important especially when users are concerned about privacy. The main difference between a WARC file generated by *Squidwarc* and one generated by *Wget* is that Headless Chrome (used by *Squidwarc*) has the ability to execute JavaScript. This ability allows for the discovery of URIs of embedded resources that otherwise would not be preserved by tools that do not execute JavaScript, like *Wget*. The command "`archivenow --wget=my.warc --ia www.example.com`" downloads the web page `example.com` using *Wget* and preserves the downloaded page in a WARC file named `example.warc`. It also sends a request to IA to capture the web page. Similarly, the option `--squidwarc` can be used for preserving the web page using *Squidwarc* to generate a WARC file.

4 CONFIGURING A NEW ARCHIVE

Even though it is currently configured to save resources into four public archives, adding a new archive can be achieved by writing a new archive handler (e.g., `{identifier}_handler.py`). For example, if we want to add a new archive named "`My Archive`", we would create a file `ma_handler.py` and store it in the folder `handlers`. The "`ma`" will be the archive identifier, so to request archiving a web page (e.g., `www.example.com`) in this new archive through Python, we would write:

```
import archivenow from archivenow
archivenow.push("www.example.com","ma")
```

In `ma_handler.py`, the class name must be `MA_handler`. In addition, this class must have at least one function `push()`, which has at least one argument for passing a URI. The second argument is optional and consists of a list of key-value pairs if required by an archive to process archiving requests (e.g., an API key for *Perma*). At least two variables should be declared in the class: a boolean variable `enabled` to toggle whether requests are sent to a specified archive and a literal `name` to specify an identifier for an archive. Removing an archive can be accomplished by (1) moving the archive handler file from the folder `handlers`, (2) renaming the archive handler file to an other name that does not end with `_handler.py`, or (3) setting the variable `enabled` inside the handler file to `False`.

5 CONCLUSIONS

We introduced *ArchiveNow* for preserving web pages in multiple on-demand web archives. In addition to requesting web archives to capture a URI, *ArchiveNow* also generates WARC files for local and private archives using *Wget* and *Squidwarc*. Furthermore, *ArchiveNow* allows customization to configure new archives or remove existing configured archives. In future work, we will continue configuring new archives whenever they become available.

6 ACKNOWLEDGEMENTS

This work is supported in part by The Andrew W. Mellon Foundation (AMF) grant 11600663 and NSF grant III 1526700.

REFERENCES

[1] Mohamed Aturban. 2017. Archivenow - A Tool To Push Web Resources Into Web Archives. https://github.com/oduwsdl/archivenow. (February 2017).
[2] John Berlin. 2017. Squidwarc - A high fidelity archival crawler that uses Chrome or Chrome Headless. https://github.com/N0taN3rd/Squidwarc. (July 2017).
[3] Free Software Foundation. 2013. GNU Wget - Introduction to GNU Wget. https://www.gnu.org/software/wget/. (2013).
[4] International Internet Preservation Consortium (IIPC). 2005. OpenWayback. https://github.com/iipc/openwayback/wiki. (October 2005).
[5] ISO 28500. 2009. WARC (Web ARChive) file format. http://www.digitalpreservation.gov/formats/fdd/fdd000236.shtml. (August 2009).
[6] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. 2014. Mink: Integrating the Live and Archived Web Viewing Experience Using Web Browsers and Memento . In *Proceedings of JCDL*. 469–470.
[7] Mat Kelly and Michele C Weigle. 2012. WARCreate: Create Wayback-Consumable WARC Files from Any Webpage. In *Proceedings of JCDL*. 437–438.
[8] Ilya Kreymer. 2013. PyWb - Web Archiving Tools for All. https://github.com/ikreymer/pywb. (December 2013).
[9] Ilya Kreymer. 2015. Webrecorder - a web archiving platform and service for all. https://webrecorder.io. (2015).
[10] Ben Welsh. 2016. PastPages. https://github.com/pastpages. (2016).