THE ALABAMA DIGITAL PRESERVATION NETWORK
PRESERVING ALABAMA'S DIGITAL RESOURCES

Getting Started with LOCKSS: A Step-by-Step Guide

By Midge Coates, Auburn University Libraries and ADPNet

Step 1: Decide what you want to preserve.

You'll need to think about what you want to preserve—the original document files, the "added value" files (transcriptions, etc.), the entire collection in its content-management software (e.g. a CONTENTdm collection), or something else.

Each Archival Unit (AU) must be Web-accessible so the LOCKSS daemon can get at it. That means that it needs to be put on a Web-accessible server computer. If you have a server, but it isn't Web-accessible or access to it is blocked (at the firewall, for example), the LOCKSS crawl won't work. You can use a firewall to protect your collection, but make sure the LOCKSS daemon has access. Check with your IT support person or system administrator to confirm that this is the case.

In the case of digital collections that use proprietary content-management systems like CONTENTdm, it is advisable to preserve the **original** image collection (that is, the digital masters or archival TIFs) with the CONTENTdm metadata. You can do this by exporting the metadata from the CONTENTdm collection (or other content management system) as a tab-delimited text file and storing this file in the same folder with your digital master files on the Web server. For this to be most effective, your collection metadata should include identifiers that match up to your original image files.

If you've made transcription files for a collection (audio, handwritten documents, etc.) using MS Word, you should probably re-save these as plain text files. Non-proprietary files are smaller and won't become obsolete.

Step 2: Write the LOCKSS Manifest Page.

The next major part of getting collections ingested is preparing each collection's manifest page. The manifest page is a basic HTML document that contains at least two things: the permission statement that gives the LOCKSS daemon the right to harvest the collection (at the foot of the document) and the base URL of the AU for the collection (or for a fraction of the collection).

Here is an example of a fairly simple manifest page for a fairly simple collection (Auburn University's Alabama Postcards images collection). Feel free to use it as a template.

Alabama Postcards Collection

Alabama Postcards files

- tifs: postcard tifs
- xml: Dublin Core metadata (minimal) for tif images, from earlier project
- notAla_jpgs: postcard jpgs, not Alabama images, origin unknown

LOCKSS

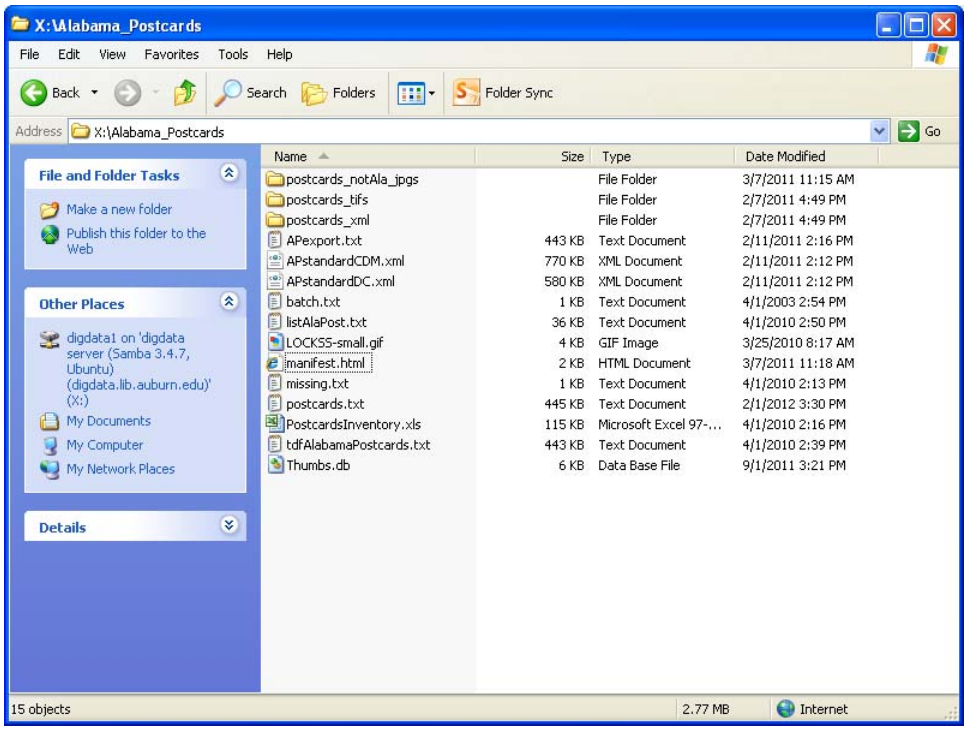
LOCKSS system has permission to collect, preserve, and serve this Archival Unit

The base URL isn't visible if you've opened the manifest in a browser window, because it's located inside the HTML <a href> tag. To see this tag, open the HTML file in DreamWeaver or Notepad++ . If you're opening it in a browser window, go to View => Source to see the tags.

I usually put all the URLs for the sub-folders in my AU, but the LOCKSS folks tell me it isn't really necessary, as the daemon will go to all the folders inside the original unless you tell it not to. So you could leave out everything except the base URL. NOTE: These URLS are **not** the CONTENTdm collection URLs; they're the URLs for the collections (or AUs) on your Web server.

The manifest page should be kept in the same folder as the AU so that the LOCKSS daemon can find it easily. If you break a collection up into digestible chunks (50 GB or so), you should have a separate manifest file for each AU chunk, with each AU chunk in a separate folder, along with its own manifest page. I also like to do a metadata export from the relevant CONTENTdm collection as a tab-delimited text file and put a copy of that in each folder also, in case the CONTENTdm collection needs to be reconstituted after a disaster.

So, if you have one AU for a collection, you need one manifest with the appropriate base URL. If you have two AUs, you need two manifests, one in each AU folder, each with the appropriate base URL for that particular AU. Here is a screenshot of the folder directory for the Alabama Postcards AU, so you can see what is in the folder. The manifest has a little box around it, so you can't miss it.



Step 3: Write the LOCKSS plugin.

The plugin is a brief set of instructions that tell the LOCKSS daemon which files to harvest from the Web server. Here is a copy of Auburn University's "generic" plugin for ingesting content into ADPNet.

```
- <map>
  - <entry>
      <string>plugin_config_props</string>
    - <list>
      - <org.lockss.daemon.ConfigParamDescr>
          <key>base_url</key>
          <displayName>Base URL</displayName>
          <description>Usually of the form http://<journal-name>.com/</description>
          <type>3</type>
          <size>40</size>
          <definitional>true</definitional>
          <defaultOnly>false</defaultOnly>
        </org.lockss.daemon.ConfigParamDescr>
      </list>
    </entry>
  - <entry>
      <string>plugin_version</string>
      <string>4</string>
    </entry>
  - <entry>
      <string>au_name</string>
      <string>"Auburn Directory Plugin, Base URL %s", base_url</string>
    </entry>
  - <entry>
      <string>au_crawl_depth</string>
      <int>99</int>
    </entry>
  - <entry>
      <string>au_start_url</string>
      <string>"%smanifest.html", base_url</string>
    </entry>
  - <entry>
      <string>au_def_new_content_crawl</string>
      <long>5356800000</long>
    </entry>
  - <entry>
      <string>au_def_pause_time</string>
      <long>6000</long>
    </entry>
  - <entry>
      <string>plugin_name</string>
      <string>Auburn Directory Plugin</string>
    </entry>
  - <entry>
      <string>plugin_identifier</string>
      <string>edu.auburn.lib.directory.AuburnDirectoryPlugin</string>
    </entry>
  - <entry>
      <string>au_crawlrules</string>
    - <list>
        <string>1,"^https?://%s/.*\.(bmp|css|gif|ico|jpe?g|js|png|tif?f)$", base_url_host</string>
        <string>4,"^%s", base_url</string>
        <string>1,"^%smanifest\.html$", base_url</string>
        <string>1,"^%s", base_url</string>
      </list>
    </entry>
</map>
```

It's pretty plain vanilla, but it's been working for us. It basically tells the LOCKSS daemon to go to the AU's base URL, find the manifest, and harvest everything found in that folder (including sub-folders). You may be able to use this as-is. Check with the LOCKSS support folks to see if it will work for your collections/AUs.

To look at the generic plugin and/or use it as a template, open it in a program (like Dreamweaver or Notepad++) that displays xml and/or html. There's an annotated version (with explanations of the various parts) on the ADPNet site at http://www.adpn.org/docs/pdf/ADPNAnnotation.pdf.

Step 4: Notify LOCKSS.

E-mail the folks at lockss-support@support.lockss.org when you have AUs ready for harvest. Include the following things in the e-mail: the AU title, the AU base URL, the estimated size of the AU, and the plugin to be used (the generic plugin or one that you created specifically for your collection). LOCKSS support staff will review your plugin and let the network know when your AU is ready to be harvested.

If you have questions about getting started with LOCKSS, send a note to the ADPNet list at adpnet@auburn.edu.

(June 2013)