THE ALABAMA DIGITAL PRESERVATION NETWORK
PRESERVING ALABAMA'S DIGITAL RESOURCES

LOCKSS Content Restoration Documentation

Jody DeRidder, University of Alabama

The following is a reconstruction (on 1 February 2010) of the process followed for restoration of content from LOCKSS cache, in March 2009.

We had 3 small collections in the original storage (at the following URL: http://libapp.lib.ua-net.ua.edu/lockss):

- Emphasis68_Recordings
- PBO_ConfederateImprints, and
- Cunningham.

The collections had been lost due to failure to communicate the location of the content to the system administrator prior to a system rebuild. Thib Guicherd-Callin of Stanford LOCKSS support provided us with the cached version of our content, from which we needed to reconstruct this repository.

A. Directory structure of cached content, and file naming

The cached content to be restored came to us in the following form.

Within the top level directory there were 3 subdirectories named for a lower-case letter: “e”, “f”, and “g”.

Each of these three subdirectories contained the following subdirectories (clearly reflecting the original URL for accessing our LOCKSS content):

/libapp.lib.ua-net.ua.edu/http/lockss/

So, thus far we have these three hierarchies of directories in the cache:

/e/libapp.lib.ua-net.ua.edu/http/lockss/
/f/libapp.lib.ua-net.ua.edu/http/lockss/
/g/libapp.lib.ua-net.ua.edu/http/lockss/

Within each of these directories were directories for a single collection.

- Emphasis68_Recordings was in:
  /e/libapp.lib.ua-net.ua.edu/http/lockss/Emphasis68_Recordings

- PBO was in:
  /f/libapp.lib.ua-net.ua.edu/http/lockss/ PBO_ConfederateImprints
- and Cunningham was in:
  /g/libapp.lib.ua-net.ua.edu/http/lockss/Cunningham

Within these directories were directories consisting of the original file names. That is to say, if the original file in Emphasis was named EMPH68_Panel_Fri_tape_2.aif, then I would find a sub-directory by the name EMPH68_Panel_Fri_tape_2.aif within the /e/libapp.lib.ua-net.ua.edu/http/lockss/Emphasis68_Recordings directory.

Within the directory named for the original file was a subdirectory named "#content" which contained a file named "current" which actually was the original file named in the directory above #content. That is to say, the actual cached file for the original file http://libapp.lib.ua-net.ua.edu/lockss/EMPH68_Panel_Fri_tape_2.aif
was this:

/e/libapp.lib.ua-net.ua.edu/http/lockss/Emphasis68_Recordings/EMPH68_Panel_Fri_tape_2.aif/#content/current

By renaming this "current" file to "EMPH68_Panel_Fri_tape_2.aif" and relocating it to its original location in the archive, I was able to restore our archive from the cache.

B. The logic behind the process of restoration

In the subdirectory of each collection:

1. Locate the subdirectory created from the item name
2. Save that item name
3. Look deeper into the #current directory for the file named "content"
4. Rename this file for the item name

C. Instructions and script for content reconstruction

Here are the original instructions and script, sent to us by Thib Guicherd-Callin on September 8, 2008:

```
"The script below should work. I'm assuming you'll know how to get it
into an executable file on libapp -- if you don't, ask me. Say you put
this script in /tmp/extract.sh and that you make it executable.

The script needs one parameter, the directory on libapp into which the
content is to be dropped. It need to correspond to what is
http://libapp.../lockss on libapp; for example /var/www/public/lockss or
something like that.
```

You need to be in the directory where you unpacked the AUs, the one that contains the three subdirectories e/, f/ and g/ (if I recall correctly). Descend into that directory then run the two commands:

```
script /tmp/transcript.txt
/tmp/extract.sh /var/www/public/lockss
```

and press Ctrl+D when the script finishes running. If there are any error messages -- which there might well be -- e-mail me the file /tmp/transcript.txt.

Thib”

```
#!/bin/sh
DSTDIR=$1
for d in *; do
   SD="$d/libapp.lib.ua-net.ua.edu/http/lockss"
   if [ ! -d $SD ]; then
     echo "error: source $SD not found"
   else
     cd $SD
     FILES=`find . -type d -exec test -f \
       {}/\#content/current \; -print`
     for f in $FILES; do
       SF="$f/#content/current"
       DF="$DSTDIR/$f"
       mkdir -p `dirname $DF`
       cp $SF $DF
     done
   fi
done
```

***The actual script had an error which is corrected here: the second “DSTDIR” reference was misspelled as “DESTDIR.”*

D.   Our alterations and problems encountered

We altered a couple of lines in this script and put in a print statement so we could watch what it was doing. Our version is as follows, with the changes highlighted:

```
#!/bin/sh
DSTDIR=$1
for d in *; do
   SD="/home/jjbattles/lockss/$d/libapp.lib.ua-net.ua.edu/http/lockss"
   if [ ! -d $SD ]; then
```

```
    echo "error: source $SD not found"
  else
    cd $SD
    FILES=`find . -type d -exec test -f \
      {}/\#content/current \; -print`
    for f in $FILES; do
      SF="$f/#content/current"
      DF="/var/www/lockss/$f"
      echo "making directory $DF\n";
      mkdir -p `dirname $DF`
      echo "copying $SF --> $DF \n";
      cp -p $SF $DF
    done
  fi
done
```

Basically, the script locates the original collection name in the directory structure of the cache, and creates a new directory where the content needs to be placed. It also locates the original filename in the directory structure of the cache, and then takes the #content/current file and renames it to the original filename, placing it in the new directory.

The process did have some problems. For one thing, we failed to exclude the directories "." and ".." in the script, which in Linux/Unix systems mean "this directory" and "the directory above this one" (respectively), so the script threw errors when it tried to overwrite files with the same thing, or tried to copy things that didn't exist anymore. This resulted in non-consequential error messages such as the following:

```
mkdir: cannot create directory
`/var/www/lockss/./Emphasis68_Recordings/EMPH68_Panel_Fri_tape_2.aif': File exists
```

and

```
error: source f/libapp.lib.ua-net.ua.edu/http/lockss not found
```

Perhaps for the same reason, the script sometimes would create "dirname" directories instead of doing what was intended, which was not helpful. Additionally, we failed to test for existence of valid values for directory and file before each copy. Spaces in filenames or directory names clearly cause problems on Linux.

However, once I understood the intent of the process (see the logic section B above) repairing the problems was simple.

E. Other files in the cache

Other files of interest in the cache were the following, located within the directories named for a single letter. These samples came from the /e/ directory:

#au_id_file
Sample content:

```
#ArchivalUnit id info
#Fri Jun 29 21:51:20 GMT 2007
au.id=edu|ua|adpn|emphasis68|Emphasis68RecordingsPlugin&base_url~http%3A%2F%2Flibapp%2Elib%2Eua-net%2Eua%2Eedu%2Flockss%2F
```

#au_state.xml
Sample content:

```
<org.lockss.state.AuState>
  <lastCrawlTime>1212465393310</lastCrawlTime>
  <lastCrawlAttempt>1219344695568</lastCrawlAttempt>
  <lastCrawlResultMsg>Can&apos;t fetch permission page</lastCrawlResultMsg>
  <lastCrawlResult>8</lastCrawlResult>
  <lastTopLevelPoll>1216040371630</lastTopLevelPoll>
  <lastPollStart>1216038594393</lastPollStart>
  <lastPollResultMsg>Complete</lastPollResultMsg>
  <lastPollResult>6</lastPollResult>
  <clockssSubscriptionStatus>0</clockssSubscriptionStatus>
  <v3Agreement>1.0</v3Agreement>
  <accessType>Subscription</accessType>
  <crawlUrls/>
</org.lockss.state.AuState><org.lockss.state.NodeStateImpl>
```

#id_agreement.xml
Sample content:

```
<list>
  <org.lockss.protocol.IdentityManager-IdentityAgreement>
    <lastAgree>1216040371583</lastAgree>
    <lastDisagree>1190282199533</lastDisagree>
    <percentAgreement>1.0</percentAgreement>
    <highestPercentAgreement>1.0</highestPercentAgreement>
    <id>TCP:[138.26.16.12]:9729</id>
  </org.lockss.protocol.IdentityManager-IdentityAgreement>
  <org.lockss.protocol.IdentityManager-IdentityAgreement>
    <lastAgree>1216040371587</lastAgree>
    <lastDisagree>1190282199505</lastDisagree>
    <percentAgreement>1.0</percentAgreement>
    <highestPercentAgreement>1.0</highestPercentAgreement>
    <id>TCP:[216.226.178.200]:9729</id>
  </org.lockss.protocol.IdentityManager-IdentityAgreement>
  <org.lockss.protocol.IdentityManager-IdentityAgreement>
    <lastAgree>1216040371607</lastAgree>
    <lastDisagree>1190282199519</lastDisagree>
    <percentAgreement>1.0</percentAgreement>
    <highestPercentAgreement>1.0</highestPercentAgreement>
    <id>TCP:[70.158.1.199]:9729</id>
  </org.lockss.protocol.IdentityManager-IdentityAgreement>
  <org.lockss.protocol.IdentityManager-IdentityAgreement>
    <lastAgree>1216040371611</lastAgree>
    <lastDisagree>1190282199526</lastDisagree>
    <percentAgreement>1.0</percentAgreement>
    <highestPercentAgreement>1.0</highestPercentAgreement>
    <id>TCP:[192.245.165.9]:9729</id>
  </org.lockss.protocol.IdentityManager-IdentityAgreement>
  <org.lockss.protocol.IdentityManager-IdentityAgreement>
    <lastAgree>1216040371592</lastAgree>
    <lastDisagree>1190282199540</lastDisagree>
    <percentAgreement>1.0</percentAgreement>
    <highestPercentAgreement>1.0</highestPercentAgreement>
    <id>TCP:[216.109.53.56]:9729</id>
  </org.lockss.protocol.IdentityManager-IdentityAgreement>
  <org.lockss.protocol.IdentityManager-IdentityAgreement>
    <lastAgree>1216040371616</lastAgree>
    <lastDisagree>0</lastDisagree>
    <percentAgreement>1.0</percentAgreement>
    <highestPercentAgreement>1.0</highestPercentAgreement>
    <id>TCP:[131.204.172.60]:9729</id>
  </org.lockss.protocol.IdentityManager-IdentityAgreement>
```

#node_props
Sample Content:

```
#Node properties
#Fri Jun 29 23:19:57 GMT 2007
node.du.size=11790870528
node.child.count=1
node.du.contentSize=11787473877
node.tree.size=11787473877
```

#nodestate.xml
Sample Content:

```
<org.lockss.state.NodeStateImpl>
  <crawlState>
```

```
    <type>1</type>
    <status>4</status>
    <startTime>1212465393310</startTime>
  </crawlState>
  <polls/>
  <activeV3Polls/>
  <completedV3Polls/>
  <hashDuration>872618</hashDuration>
  <curState>0</curState>
```

There were also similar #something files at the file level where #current existed. One of those contained the checksum for the file. Another contained the original file path to the location of the file. There may have been others.

We deleted those files, so I do not have a record of them.