Retrieving HathiTrust Research Center Open-Open Corpus

04 October 2013

The HTRC Open-Open corpus of non-Google digitized content is accessible programmatically through the HTRC Sandbox. This document contains instructions for retrieving chunks of the HTRC open-open corpus programmatically. To obtain a copy of the volume list zip file, contact HTRC at htrctechhelp@gmail.com

List of HTRC volume IDs

The HTRC open-open corpus contains 255,176 volumes with each volume identified by a **volume ID.** The **volume ID is** used to fetch the volume through HTRC data API.

Volumes can be retrieved from the HTRC Sandbox Open-Open corpus programmatically through the HTRC Data API, providing it a list of volume IDs. The HTRC Data API then returns an output zip file of results for each call. HTRC has a list of all Volume IDs for the Open-Open corpus that it can share. This zipped file, called *id-split.zip,* splits the volume IDs in 13 files, with the first 12 having 20,000 volume IDs and the 13th having the remaining 15,176 volume IDs.

Run Python script DownloadVolumes.py

The Python script DownloadVolumes.py can be used to download volumes. The script, available at http://wiki.htrc.illinois.edu/display/COM/Python+client+for+HTRC+Data+API , requires two arguments: volume list file containing a list of volume IDs, and path for the output as a zip file. The script has the location of the HTRC Sandbox Data API service address written into it. The script can be run as follows:

```
python DownloadVolumes.py <volume-ids-file> <output-zipfile>
```

For example, if the volume-ids-file is vid_splitaa (which is one of the volume lists in the *id-split.zip*) and the output zip file is named as output_aa.zip, then the command is:

```
python DownloadVolumes.py vid_splitaa output_aa.zip
```

After unzipping the output zip file, a folder exists for each volume in the volume list, and within that, the text of the volume.

```
loc.ark+=13960=t2s47863z                    <= volume ID
         | 00000001.txt                     <= page
         | 00000002.txt
         |…
uc2.ark+=13960=t9w094g4m
         | 00000001.txt
         | 00000002.txt
         |…
```

The Python client was developed in Python 2.7, and it is recommended to run it in Python 2.7.5 environment.

Script Parameters

The parameters in the python script are in a good default mode, but what follows lists the parameters and other configuration options:

- VOLUME_PARAMETERS. Three options exist; uncomment the option you wish to use; leave others commented.
    - VOLUME_PARAMETERS = {}

      Empty: returns only volume content
    - VOLUME_PARAMETERS = {'mets' : 'true'}

      Returns METS record along with the volume content.
    - VOLUME_PARAMETERS = {'mets' : 'true', 'concat' : 'true'}

      Returns METS record, volume content AND concatenates all the pages into one single text file per volume.
- BATCH_SIZE
    - BATCH_SIZE = 50

      This is the batch size, which is the number of volumes returned each time. It is recommended that batch size 50 is used for best performance.

Notes to Mac Users

The output zip file generated by the Python client is in Zip64 format which supports larger than 4G zipped file. You need to use a proper unzip utility to unzip the zipped file. For Mac user, it is likely that the unzip utility shipped with the OS does not support Zip64. You might get an error looking like:

"error [X.zip]: start of central directory not found;
zipfile corrupt.
(please check that you have transferred or created the zipfile in the appropriate BINARY mode and that you have compiled UnZip properly)".

You can read more from this link.
http://superuser.com/questions/114011/extract-large-zip-file-50-gb-on-mac-os-x

There are two ways to work around the unzip error caused by Zip64 format:
1) Install an unzip utility (e.g. unzip 6.0) which can handle zip64 file.
2) Chop the volume ID file into small chunks, e.g. 5000 volume IDs, to return a zipped file under 4G.