

Ninth Session

Joshua Ng • Archives New Zealand
Reto Kromer • AV Preservation by reto.ch

Bash Scripting for Audio-Visual Preservation
webinar series hosted by AMIA
22 February 2022

1

Contents

- data maintenance at data migration
- summary of the scripts
- coda

2

Migration

3

read | script | write

script to modify

- container
- codec
- both container and codec
- metadata
- filename

4

Video transcoding

from:

- AVI / 8-bit or 10-bit uncompressed
- QuickTime (.mov) / 8-bit or 10-bit uncompressed
- MP4 / 8-bit or 10-bit uncompressed

into:

- Matroska (.mkv) / FFV1

5

Container and video codec

- read archived file from source LTO
- demultiplex from source container format
- decode using source video codec
 - $Y'CbCr$, 4:2:2, 8-bit or 10-bit "raw"
- encode using new video codec
- multiplex into new container format
- write new file to destination LTO

6

Reading

Reto Kromer: **On the Bright Side of Data Migrations**, in «IASA Journal», n. 49 (December 2018), IASA, p. 18–22

→ retokromer.ch/publications/IASA_49

7

Summary

8

File generation scripts

- make a checksum manifest
- make an access file
- make a mezzanine file

9

Data maintenance scripts

- find the missing items in a folder of numbered files
- verify a Baglt File Packaging, according to RFC 8493
- update a checksum manifest

10

Add or delete files (1)

add files to or delete files from a folder

- modify folder
- update manifest

11

Add or delete files (2)

add files to or delete files from a Baglt archive

- modify data/
- update manifest-<algorithm>.txt
- update tagmanifest-<algorithm>.txt

12

Modify Baglt version

modify the version of a Baglt archive

- modify bagit.txt file
- update tagmanifest-<algorithm>.txt

13

Change Baglt algorithm

change the checksum algorithm of a Baglt archive

- generate new manifest-<algorithm>.txt
- update tagmanifest-<algorithm>.txt

14

Coda

15

Install and uninstall scripts

to install

```
cp script.sh /usr/local/bin/script
```

to uninstall

```
rm -i /usr/local/bin/script
```

16

```
# chain successive scripts

while script in "${script_list}"; do
  ${script} "${parameters[@]}"
  if (( $? != 0 )); then
    report_error
  else
    report_success
  fi
done
```

17

```
# batch process successive jobs

while IFS= read -r job; do
  ${job}
  if (( $? != 0 )); then
    report_error
  else
    report_success
  fi
done < "${job_list}"
```

18

```
# batch process successive script chains

while IFS= read -r job; do
  while script in "${script_list["${job}"]}"; do
    ${script} "${param[@]}" &>/dev/null & wait $!
    if (( $? != 0 )); then
      report_error
      break
    else
      report_success
    fi
  done
done < "${job_list}"
```

19

||: Coda :||

20

Play a script as a video

```
ffplay script.txt
```

```
ffplay -loop 0 script.txt
```

21

A last one-liner

```
(cd ${HOME}/Desktop; curl -O  
https://reto.ch/training/  
2022/202202/script_{7,9}.sh >  
script_9.sh; chmod +x  
script_{7,9}.sh)
```

```
curl https://reto.ch/training/  
2022/202202/commands_9.txt | bash
```

22

AV Preservation by
reto.ch

zone industrielle Le Trési 3
1028 Préverenges
Switzerland

Web: reto.ch
Twitter: [@retoch](https://twitter.com/retoch)
Email: info@reto.ch



23