Automate Metadata Checks

This How-to article is intended to provide all the relevant information required to automate metadata checks. Metadata checks are necessary for testing purposes and are currently a part of roughly 70% test cases. This set of scripts intend to target reducing test duration as well as enhance ease-of-testing by automating redundant tester responsibilities. Potentially, this set of scripts might find usability for the development team as well as become a part of a Fully-Automated Test Case/Suite.

Instructions

Step-by-Step Guide;

1. Copy Scripts folder to H1 Ubuntu-Linux shell
    - a. Download the Scripts package, unzip(if required) and move it to User home folder. **NOTE:** Place the Scripts folder in the Home directory ONLY. The Script paths are relative to Home directory and might not work when source from another directory.
        - i. **cd ~/**
        - ii. **tar -xvf Scripts.tar**
    - b. Check contents of the scripts folder to confirm.
        - i. **ls Scripts/**
2. Set environment variables.
    - a. Set Location Variable to the relevant folder H1 is storing the files(folder changes when encryption option is selected).
        - i. **vim Scripts/cam1_files.sh**
        - ii. **LOC='/media/ubuntu/USB/cobanvideos**'
        - iii. **vim Scripts/Cam1_metadata.sh**
        - iv. **LOC='/media/ubuntu/USB/cobanvideos**'
3. Source Scripts.
    - a. Trigger a recording.
    - b. To just view the files associated with the recording triggered:
        - i. **Source Scripts/cam1_files.sh**
    - c. To view entire metadata being generated as the operation is performed:
        - i. **Source Scripts/Cam1_metadata.sh**

Usage:

1. Source Scripts one Camera at a Time:
    - a. Trigger a recording.
    - b. To just view the files associated with the recording triggered:
        - i. **Source Scripts/cam1_files.sh**
    - c. To view entire metadata being generated as the operation is performed:
        - i. **Source Scripts/Cam1_metadata.sh**

        > **Note**
        > There are Similar scripts for Camera_2, Camera_3, and Snapshot metadata which are only different in the label title. For Ex. Above script is for Camera 1 metadata. We will source "**cam2_files.sh**" and "**Cam1_metadata.sh" for Camera_2** and so on for Camera 3.

2. Source Scripts 3 Camera-recordings at a Time:
    - a. Trigger a recording.
    - b. To view entire metadata being generated as the operation is performed:
        - i. **Source Scripts/3v.sh**
3. Source Scripts for latest Snapshot at a Time:
    - a. **Source Scripts/fetch_snap.sh**

Automating Metadata Fetching Using CRON functionality (CRON Jobs):

1. **To edit or create your own crontab file, type the following command at the UNIX / Linux shell prompt:**

    ```
    $ crontab -e
    ```

    The syntax is:

    ```
    1 2 3 4 5 /path/to
    /command arg1 arg2
    ```

OR

```
1 2 3 4 5 /root
/backup.sh
```

Where,

- 1: Minute (0-59)
- 2: Hours (0-23)
- 3: Day (0-31)
- 4: Month (0-12 [12 == December])
- 5: Day of the week(0-7 [7 or 0 == sunday])
- /path/to/command – Script or command name to schedule

2. Place the following command in your crontab:
   0-59 * * * * /home/ubuntu/croncom.sh > /home/ubuntu/Metadata/gsnext_metadata.txt 2>&1
3. In croncom.sh , the user can have commands that need to be run automatically every "x" minutes/hours/days etc. such as:

```
$ #!/bin/bash
source /home/ubuntu/Scripts/3v.sh > /home/ubuntu/Metadata/gsnext_metadata.txt
source /home/ubuntu/Scripts/3v.sh > /home/ubuntu/Metadata/gsnext1_metadata.txt
```

Automating IPC Command Triggers on H1 using Shell Scripts

1. **Multiple IPC Commands from the following sources can be grouped and placed in a script to test the whole set at once:**
   a. **IPC Implementation**
   b. **Testing**

For Ex.:

```
!/bin/bash
mosquitto_pub -u CobanH1 -P H1Refactor -t "MetaManager/STS/TRIGGER/42" -m "{\""CODE\"":\""42\"", \""TIME\"":\""0\""}"
source /home/ubuntu/new_Scripts/wait.sh
mosquitto_pub -u CobanH1 -P H1Refactor -t "MetaManager/STS/TRIGGER/43" -m "{\""CODE\"":\""43\"", \""TIME\"":\""0\""}"
source /home/ubuntu/new_Scripts/long_wait.sh
mosquitto_pub -u CobanH1 -P H1Refactor -t "MetaManager/STS/TRIGGER/44" -m "{\""CODE\"":\""44\"", \""TIME\"":\""0\""}"
```

The metadata scripts come in handy to verify the IPC trigger-based recordings have the expected metadata.

Automating Video/Audio Quality Checks Using CRON functionality (CRON Jobs):

Installing FFMPEG libraries:

1. **sudo apt-update**
2. **sudo apt-get update**
3. **sudo apt install ffmpeg**

Placing command inside crontab

1. Place the following command in your crontab:
   0-59 * * * * /home/ubuntu/croneod.sh > /home/ubuntu/Metadata/gsnext_metadata.txt 2>&1
2. In croneod.sh , the user can have commands that need to be run automatically every "x" minutes/hours/days etc. such as:

```
#!/bin/bash
source /home/ubuntu/new_Scripts/vid2_check.sh > /home/ubuntu/Metadata/vid_check.txt
```

3. **A useful vid_check.sh script example is following:**

```
LOC='/media/ubuntu/USB/cobanvideos/'
ls -latr $LOC | grep '\.mp4' | tail -6 | awk -v N=$9 '{print $9}' | cut -d '.' -f1 > /home/ubuntu/new_Scripts/mp4_files.txt
#split -l 1 5mp4.txt
mapfile -t array < /home/ubuntu/new_Scripts/mp4_files.txt
file0=${array[0]}
file1=${array[1]}
file2=${array[2]}
file3=${array[3]}
```

```
file4=${array[4]}
file5=${array[5]}
form_v=".v"
form_mp4=".mp4"
form_a=".a"
form_ok=".ok"
form_l=".l"

echo $file0$form_mp4
echo $LOC$file0$form_mp4
vv0=$(ls -latr $LOC | grep '\-0.v' | tail -1)
echo $vv0 | awk -v N=$9 '{print $9}'

/usr/bin/ffprobe -i $LOC$file0$form_mp4 -show_streams -select_streams a -loglevel error
```

4. Sample Result inside **/home/ubuntu/Metadata/vid_check.txt:/media/ubuntu/USB/cobanvideos/8066@20200626065151-1.mp4**

```
8066@20200626074243-0.v
[STREAM]
index=1
codec_name=aac
codec_long_name=AAC (Advanced Audio Coding)
profile=LC
codec_type=audio
codec_time_base=1/44100
codec_tag_string=mp4a
codec_tag=0x6134706d
sample_fmt=fltp
sample_rate=44100
channels=2
channel_layout=stereo
bits_per_sample=0
id=N/A
r_frame_rate=0/0
avg_frame_rate=0/0
time_base=1/44100
start_pts=441
start_time=0.010000
duration_ts=28727279
duration=651.412222
bit_rate=196463
max_bit_rate=200000
bits_per_raw_sample=N/A
```

Related articles

- Automate Metadata Checks
- Using H1 Cross-Compile VM Clone
- H1 Screenshot
- Qt 5.12.4 - Qt Cross-Compile H1 Application (In Progress)
- Development Procedure