Nearly 1 million moving image assets digitised and catalogued in the BFI Collections Information Database (CID)

- Nearly 650,000 of those are off-air TV
- Another 800,000 preserved but still to be catalogued in CID
- BFI is the National Television Archive designated by OFCOM under Provision of Broadcasting Act, 1990
• Started recording off-air TV to one inch video tape in 1985
• Teams worked around the clock to capture select broadcast shows
• 2015 off-air recording was automated with BFI fork of BBC’s Redux project
• BBC Redux project closed May 2022
• BFI launched an R&D project in 2021 resulting in STORA
• STORA modelled on Redux to avoid breaks in existing automated workflows
Astra satellites broadcast across Europe

Passed through Quatro Low Noise Block then TBS TV PCI receiver cards

Signals routed through patch fields to a multi-switch to select bands & polarisation

STORA has 3 multi-switches allowing for up to 24 different multiplexes

Cesbo Astra app demultiplexes each channel's MPEG-TS into Single Program Transport Stream

Creates Unicast RTP stream and Unicast UDP stream, both needed by STORA
- Modelling BBC Redux folder structures
- File formatting and video standard
- Folders carry metadata for date, channel, time of broadcast, duration and EventID
- Contents of folder include info.csv, a CSV format document containing programme information from the stream metadata
- Captured video is not re-encoded but raw PES data wrapped video, audio, subtitles and information tables
- Extracted transcript of the spoken word of the programmed saved to WEBVTT
- All this data critical to preservation goals
- Modelling BBC Redux folder structures
- File formatting and video standard
- Folders carry metadata for date, channel, time of broadcast, duration and EventID
- Contents of folder include info.csv, a CSV format document containing programme information from the stream metadata
- Captured video is not re-encoded but raw PES data wrapped video, audio, subtitles and information tables
- Extracted transcript of the spoken word of the programmed saved to WEBVTT
- All this data critical to preservation goals
• Modelling BBC Redux folder structures
• Folders carry metadata for date, channel, time of broadcast, duration and EventID
• Contents of folder include info.csv, a CSV format document containing programme information from the stream metadata
• Captured video is not re-encoded but raw PES data wrapped video, audio, subtitles and information tables
• Extracted transcript of the spoken word of the programmed saved to WEBVTT
• All this data critical to preservation goals
System for Television Off-air Recording and Archiving

Nagios®

CCExtractor

VLC MEDIA PLAYER

FFmpeg

libdvbtee

Media Info
Two separate approaches to recording the RTP stream originally in separate script but now combined

Electronic Programme Guide (EPG) data retrieved from commercial supplier via their Rest API

EPG data converted to JSON schedule for a single day's programmes, per channel

Script run once per day launching around midnight and runs until all scheduled items have concluded

Script relaunched immediately when finished by shell scripts that monitor for scripts that have completed running

Script indebted to ActiveNation script written in 2015
Two separate approaches to recording the RTP stream originally in separate script but now combined

Electronic Programme Guide (EPG) data retrieved from commercial supplier via their Rest API

EPG data converted to JSON schedule for a single days programmes, per channel

Script run once per day launching around midnight and runs until all scheduled items have concluded

Script relaunched immediately when finished by shell scripts that monitor for scripts that have completed running

Script indebted to ActiveNation script written in 2015
• Second script started monitoring the UDP stream service information data for change to the EventID

• Script stores the last five programme EventIDs in a list, when a new number appears it triggers new recording

• Script loops indefinitely monitoring this UDP data, creating and placing programme recordings in unique folder paths

• LIBDVBTII project makes this approach possible, and supports parsing of UDP streams

• Shell command is spawned from Python subprocess module and the response is collected and converted into dictionary
Second script started monitoring the UDP stream service information data for change to the EventID

Script stores the last five programme EventIDs in a list when a new number appears it triggers new recording

Script loops indefinitely monitoring this UDP data, creating and placing programme recordings in unique folder paths

LIBDVTEE project makes this approach possible, and supports parsing of UDP streams

Shell command is spawned from Python subprocess module and the response is collected and converted into dictionary
RTP STREAM RECORDED BY VLC

UDP STREAM READ BY LIBDVBTEE

RTP STREAM RECORDED BY VLC
The VLC python bindings create a VLC Instance and media player object

They are called in the main script to stop a completed recording and start a new recording

We use the demux dump command in the instance from VLC’s demux library, saving the stream without decoding

The append flag is used to ensure breaks in stream recordings are appended to existing files, not overwritten

Restart warning text file is automatically written to folder containing programme recording to indicate the break

```python
for key, val in running_items):
    if key not in event_list:
        # Add current eventId to event_list
        event_list.append(key)
        time_print("New running EventId: ", key), False
        time_print("Event list updated: ", event_list), False
        prog_info = val.split(', ')

        # Initialise recording path - needs date paths adding
        outfile = initialise_ts_ts(prog_info[1], key, prog_info[0])

        if len(event_list) > 1:
            # Stop existing recording
            time_print("Ending recording for previous programme", False)
            player.stop() # Stop playback
            player.release() # Close the player
            inst.release() # Destroy the instance
            inst.print("STOP Instance: ", inst), Player: ", player), Media: ", media)

        # Start new recording using initialised outfile as destination
        time_print("Initialising recording for path: ", outfile)
        [inst, player, media] = record_stream(infile, outfile)
        player.play()
        inst.print("START Instance: ", inst), Player: ", player), Media: ", media)
        inst.print("Started recording: ", prog_info[1]), (CHANNEL)

        def record_stream(instream, outfile):
            ""
            Record the network stream to the output file.
            Create VLC instance that launches demux dump and
            append to stream (if already exists) or creates new
            ""
            inst = vlc.Instance("--demux=dump", f"--demuxdump-file={outfile}", 
                "--demuxdump-append")
            player = inst.media_player_new()
            media = inst.media_new(instream)
            media.get_mrl()
            player.set_media(media)
            return (inst, player, media)
```
The info.csv file is created from metadata retrieved from Media Area software MediaInfo.

Python subprocess spawns shell to retrieve metadata of recent completed recording.

The data is converted to a list and written to a new CSV file using Python CSV module.

Similarly the subtitles.vtt file is created using from subtitles extracted from stream using CCExtractor.

Similarly the subtitles.vtt file is created using from subtitles extracted from stream using CCExtractor.

Python subprocess spawns shell to retrieve subtitles.

CCExtractor formats the file for WEBVTT which is imported to CID database as searchable text.
The info.csv file is created from metadata retrieved from Media Area software MediaInfo.

Python subprocess spawns shell to retrieve metadata of recent completed recording.

The data is converted to a list and written to a new CSV file using Python CSV module.

Similarly the subtitles.vtt file is created using from subtitles extracted from stream using CCExtractor.

Similarly the subtitles.vtt file is created using from subtitles extracted from stream using CCExtractor.

Python subprocess spawns shell to retrieve subtitles.

CCExtractor formats the file for WEBVTT which is imported to CID database as searchable text.
Nagios is an event monitoring system that issues alerts when problems are detected.

We have two forms of alert for STORA:

- Recording checks monitor current channel stream.mpeg.ts files by creating a checksum waiting four seconds and making another and comparing to ensure they are different.

- Monitoring of Cesbo Astra software to ensure each channels has status ‘onair = true’.

If either fail Nagios software shows a critical status for the channel and sends an email confirming which problem has raised the alarm.
Nagios is an event monitoring system that issues alerts when problems are detected.

We have two forms of alert for STORA.

Recording checks monitor current channel stream.mpeg.ts files by creating a checksum waiting four seconds and making another and comparing to ensure they are different.

Monitoring of Cesbo Astra software to ensure each channel has status ‘onair = true’.

If either fail Nagios software shows a critical status for the channel and sends an email confirming which problem has raised the alarm.
STORA: System for Television Off-air Recording and Archiving

The scripts in this repository form the off-air TV recording codebase responsible for preserving 17 UK Television channels 24 hours a day, 7 day a week. The BFI is the body designated by Ofcom (UK communications regulator) as the National Television Archive, under the provision in the Broadcasting Act, 1990. This designation allows us to record, preserve and make accessible TV off-air under section 75 (recordings for archival purposes) of the Copyright, Designs and Patents Act, 1988 and later the Copyright and Rights in Performance Regulations 2014 (under Research, Education, Libraries and Archives).

Overview

These scripts manage the recording of live television, accessing FreeSat streams using RTP streams for the

https://github.com/bfidatadigipres/STORA/
Thank you!

joanna.white@bfi.org.uk
www.bfi.org.uk/bfi-national-archive
github.com/bfidatadigipres/STORA