BBC

R&D White Paper

WHP 137

May 2006

A simple media logger using aaf

R. Storey

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper WHP 137

A Simple Media Logger using aaf

R Storey

Abstract

This paper describes a simple media logger. The purpose behind its development was to test the aaf Edit Protocol[1] for implementation of the logging phase of media production.

This application is one of a set of applications that implements the Ingest, Logging and editing stages of media production using aaf to communicate the results of each stage.

The application accepts ingested files from any type of content and attempts to build an aaf composition that describes the content and its derivation from tapes, films, import or recording. The media could be from a studio shoot, or perhaps from a completed programme, parts of which are candidates for re-used.

It has facilities for automatically detecting the shots in the ingested media, adjusting, inserting, deleting and re-naming shots, commenting them with shot and timed comments, setting key frames that represent their content and displaying timecode break information that was captured during the ingest. It also allows insertion of derivation information if this was not done at the ingest stage.

The application can either open an existing aaf file from an ingest session or delivered by a Media Asset Management system, or make a new one when the media is already in existence through an alternative route. It stores its results as production metadata in the same aaf file, in a form suitable for breaking the source media up into individual media assets; one definition of a media asset is 'the smallest piece of media that can be used on its own' and in this context, it is a shot.

The aaf input and output files contain all of the production metadata required and can be either stored intact by a Media Asset Management system or used to communicate between the MAM and an application. This allows integration of applications from different vendors using only the royalty free Edit Protocol and aaf SDK.

Additional key words: AAF, MXF, Edit Protocol, Ingest, Logging, Editing, integration, joined up broadcasting.

[1] www.sourceforge.net/projects/aaf

BBC Research & Development
White Paper WHP 137

A Simple Media Logger using aaf

R Storey

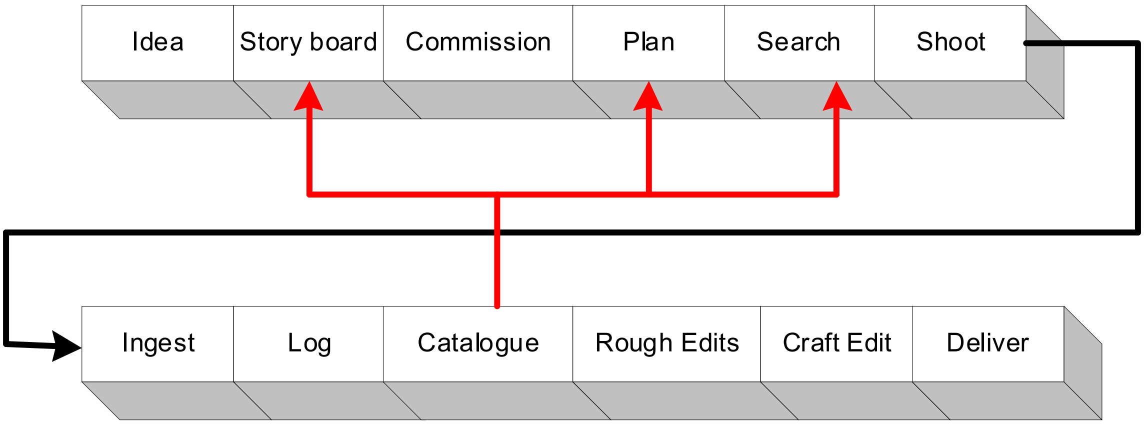
BBC Research & Development
White Paper WHP 137

A Simple Media Logger using aaf

R Storey

1 Introduction

Production processes for different types of content are, not surprisingly, different - often for very good reasons. A project has to start somewhere, so we start from a generic set of processes arranged in a reasonably efficient order in the hope that it will be able to produce content not only for linear programmes but also for web-based and 'interactive' access. I put interactive in inverted commas because real interactivity requires programmes to be manufactured as 3d objects and rendered to 2d only for those formats that cannot handle the third dimension. Currently this includes all media types produced by the major media companies but almost none of the media types used by anyone under the age of twenty. Although the production metadata provided by aaf is comprehensive, adding the third dimension is another five years work at least.



The major change form current production practice is moving cataloguing up stream so that it happens on the ingested material rather than on the finished programmes. The hope is that of the 49/50ths[2] of material that currently gets discarded, 2 or 3/50ths might be of good enough quality to be put somewhere really safe and catalogued well enough for a production researcher, armed with a reasonably competent search engine, to find it again. This neatly side steps the cost of re-shooting it and provides a reasonably searchable catalogue for the programme planning people to use. It may also be that access to a good back catalogue will help with Story Boarding and planning, and possible help create some new ideas.

Cataloguing is best done by a trained archivist, because an archivist would enjoy the job whereas the production team, most likely, would not. Cataloguing, however, requires the material to be packaged and delivered to the archivist, which is where the logging application comes in.

Logging takes ingested material and breaks it up into manageable sections. If it is done by the production team they should only log the content they want to use, leaving the remainder for the archivist. Cataloguing, however, requires creation of data according to a recognised archiving standard, which first means you have to agree what standard to use. AAF provides a means to package this information as XML so we do not need to concern ourselves with this part of the process. This also means that the logging application will not mean much to an archivist – it is a technology demonstrator of a tool for the production team.

[2] Shooting ratios vary a lot, 10:1 is exceptional, 30-50:1 is typical and some productions are nearer 2000:1.

Ingest appears somewhere in the middle of the new production processes, after commissioning, planning, archive search and shooting, but before logging, cataloguing and the production team starting work on the audio and video footage. The assumption here is that - if the material is captured on videotape and played into the ingest system using a videotape replay machine - the ingest machine will automatically record timecode breaks.

2 A simple method for joining up the programme chain

The proposal behind this work is to join up the programme chain using aaf files that point to external essence files and simple operating system functions that are in common use. The essence files should be in a published format and accessible by any application. Mxf wrapped mpeg and audio files, are used by the application but it can also read m2v and wav or multiplexed mpg files.

The operating system functions are simple directory navigation and email attachments. The idea is that when a piece of media needs to be logged, the MAM assembles an aaf file containing the piece of work and mails it to the assigned archivist. The archivist clicks the attachment which fires up the logging application and links into the media automatically using a URL. When the job is done, the archivist saves the file and returns it to the MAM which then assimilates its contents. The application is also set up to hijack the parameter string and load a new file when the user clicks a second file, without first closing the logging application.

The strengths of this approach are that it uses the same user actions as say, editing a Word document sent as an attachment. Returning it to the MAM is equivalent to emailing the modified file back to the sender. It requires no ‘middleware’ training.

The application uses a parameter string to load its files. Windows delivers a parameter string to the application assigned to a file extension when that type of file is clicked. The file contents are described in the Edit Protocol document so vendors do not need to talk to one another or a systems integrator to build a perfectly functional application. This should make joining up the programme chain a good deal more feasible than using a deep integration approach. Creating the Media Asset Management system of course, remains a challenge, but should be feasible -provided users can make up their minds what they want, commit to its development and manage the project sensibly.

3 Logging requirements

The requirements for the Logging application are derived from the proposed operating environment and the uses to which the resulting logged file will be put – either splitting an ingested set of shots into separate media assets or using the file as a Bin for loading its shots into an editor.

3.1 Creation of a logging project

The logging application follows ingest, or import if the material has already been captured using an application that is not part of a joined up broadcast chain. The application therefore needs to be able either to import an aaf file, created by a logging application or a Media Asset Management system to represent the output of an ingest system, or allow creation of a new file from ingested material, given the additional metadata required to document its derivation.

To support automated linking from an email attachment, it needs to be able to accept the parameter string delivered to the application instance by the operating system and open the file contained in it. It also needs to be able to intercept a second attempt to launch if the user fails to save the current work and shut down the logging application. Under these circumstances it needs to query whether the user has completed the current work and only if the answer is yes, should it open the new file.

The application should also be able to re-open its own files so that work can be done at different times. It is helpful if the application saves its files to the same location since this simplifies usage.

Once the work is completed the file can be saved and simply mailed back to the MAM, which will then assimilate it as a completed task into whatever structure it uses.

3.2 Breaking the media into shots.

If there is an aaf file that describes the ingest, this may or may not have been broken into shots by the ingest system so the logging application must also provide the ability to manually and/or automatically break the material up into its constituent shots. Alternatively, if an earlier shot change detection process has made mistakes, then the application must provide a means to correct the errors either by inserting missed shots, removing wrongly detected shots or by adjusting their positions. Given that automated shot change detection is a time consuming process, it should run on a separate thread so that logging can begin while the detection is in progress.

3.3 Commenting the shots

The application should provide a means to view and edit existing comments and attach new comments to the shots. Comments can be either a fixed comment for each shot or timed comments that describes particular frames within a shot. The comments should be editable and removable.

3.4 Key frame insertion

Another key item throughout later stages of the production process is key frames. These are frames that are in some way represent the content of a shot and there is one per shot. They are used in lists of clips and sometimes in timelines to remind production staff what the shot is about in a visual rather than a textual way. A logging application should allow choices to be made about which frame to use as a Key frame and allow a comment to be attached to the key frame.

3.5 Timecode breaks

Video tape machines store two kinds of timecode, LTC which is recorded along the tape and VTC which is recorded as part of the video signal. LTC is reliable while the tape is playing or spooling quickly whereas VTC is reliable when the tape is playing of spooling slowly. These would have the same values in an idea world but often do not. Sometimes the hours digits of one or the other timecode are used as a tape number for instance.

Often, the timecode is broken in that it either has repeated sections, breaks in the recorded time or is missing altogether.

The important thing is that the production team must be able to trace back from a frame in a production to a frame in the video tape reliably. This is essential if they are to be sure whose content they have used and who needs to be paid royalties. This requires an audit trail of the timecode on the tape which must be captured during ingest.

The production system, however, must use a system timecode which increases monotonically,. An aaf file provides a means to link a system timecode to an off-tape timecode so a logging application can relate System timecode to the off-tape timecode, which may often be seriously broken. The file format therefore provides the necessary audit trail to allow a user to navigate backwards from the frames in a production to the frame on the tape, regardless of how broken the tapes timecode is (the tape timecode must of course be present in some shape or form - aaf cannot perform miracles).

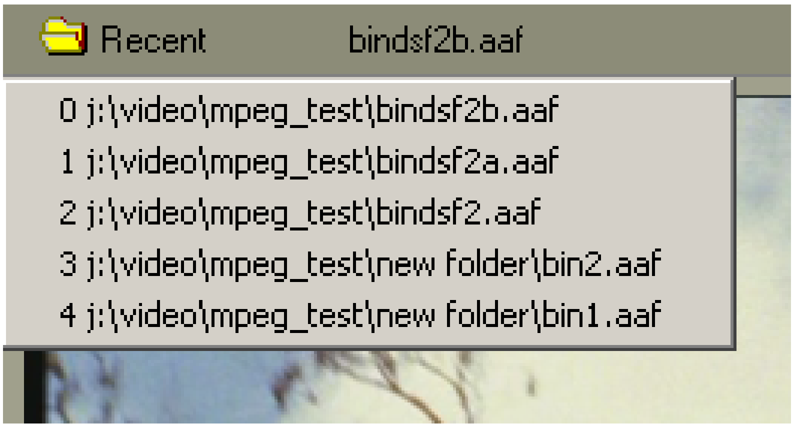
3.6 Media Replay

The logging application should be able to link to external media files specified by a Network Locator in the aaf file. These media files should be fully legal Mxf files which means the Mxf wrapper conforms to the Mxf standard and the media format is standardised. This make them potentially readable by any application. The criteria for purchasing the application in the first place is that it can access Mxf Files. This sounds like a trivial requirement but many media formats are not standardised and software applications may not replay them correctly.

4 The Logging Application

A graphic of the logging application with logging in progress is shown on the following page. The application does not have the standard menu bar but has large buttons to allow common functions to be accessed with the minimum of mouse movements, or through keyboard shortcuts which allow operations to be done more quickly.

4.1 Top Panel controls



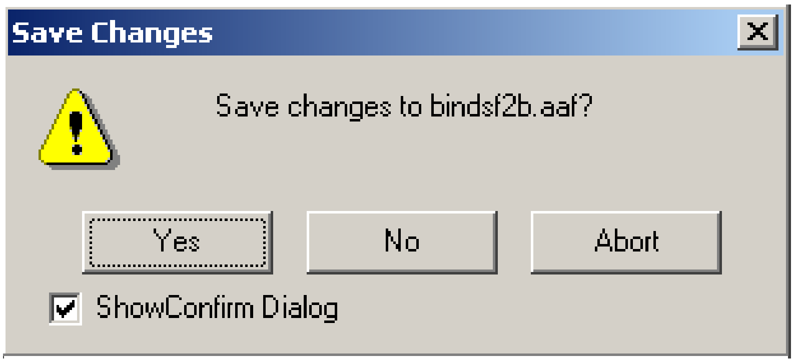
bindsf2b.aaf

0 j:\video\mpeg_test\bindsf2b.aaf
1 j:\video\mpeg_test\bindsf2a.aaf
2 j:\video\mpeg_test\bindsf2.aaf
3 j:\video\mpeg_test\new folder\bin2.aaf
4 j:\video\mpeg_test\new folder\bin1.aaf

The Recent control allows the user to see a list of recently opened files. The list shows the recently opened files in the order that they were opened and holds no surprises.

The list will not show files that have been deleted. The user can change the currently opened file for one from the list by clicking the icon or using the up and down arrow keys followed by Enter.

If the current file has no unsaved changes the new file will be opened.

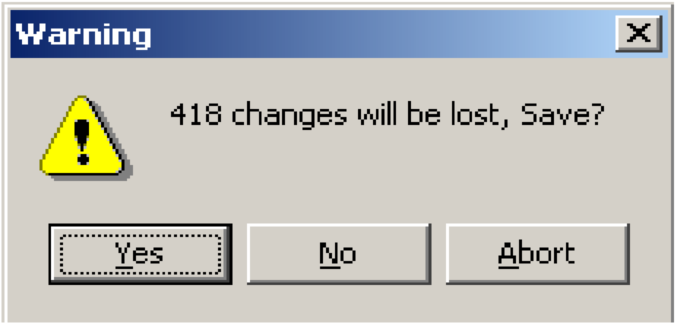


Save changes to bindsf2b.aaf?

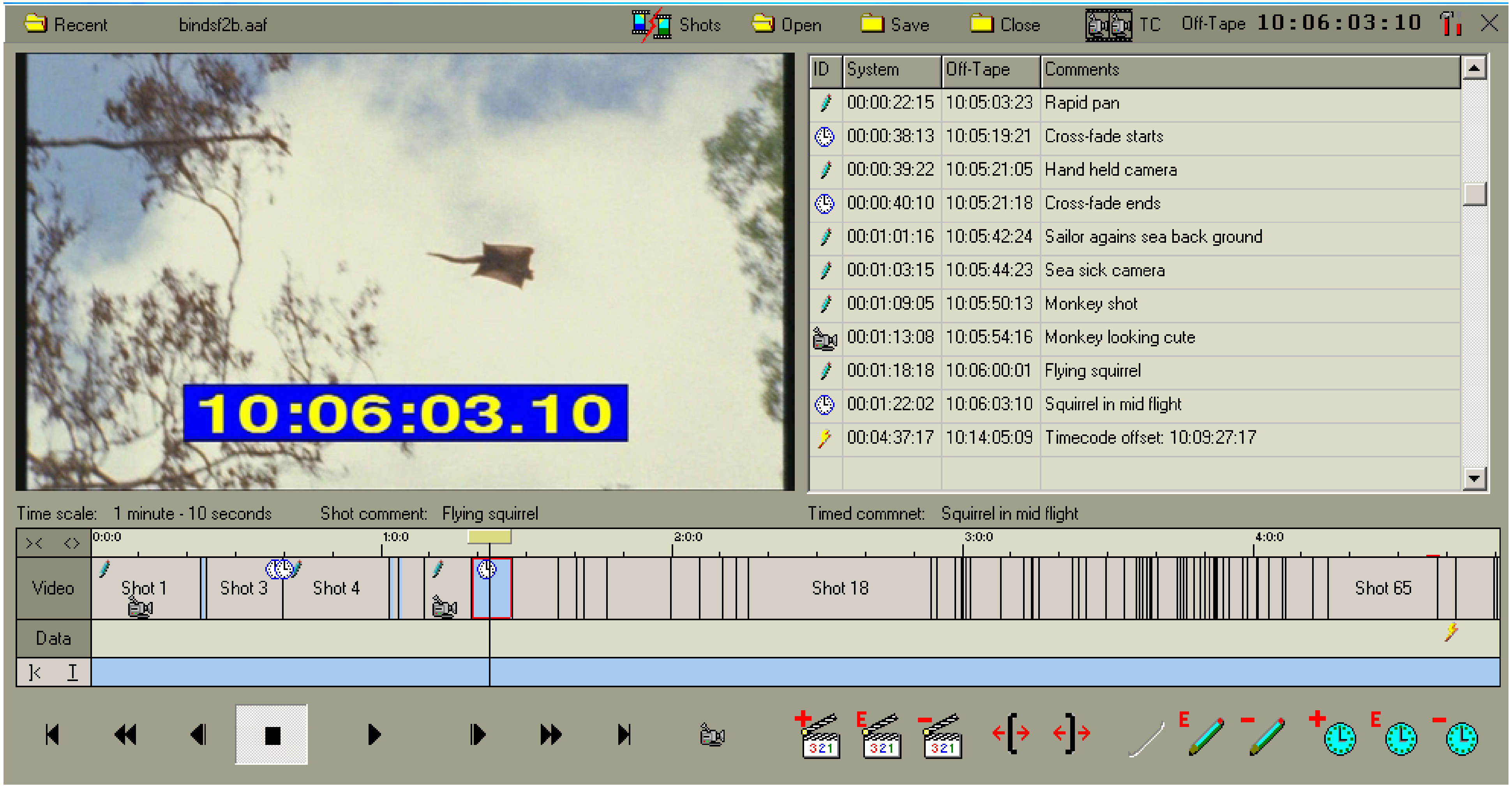
☑ ShowConfirm Dialog

If the current file has outstanding changes that have not been saved, a warning dialog will be shown.

The warning dialog has some safety features built in. If the user chooses not to save a file that has outstanding changes it will popup a second dialog warning how much work will be lost if the file is not saved. This will help to avoid loosing work when a user is tired at the end of a long day, and is a pattern used throughout the application.



418 changes will be lost, Save?



ID	System	Off-Tape	Comments
✎	00:00:22:15	10:05:03:23	Rapid pan
🕓	00:00:38:13	10:05:19:21	Cross-fade starts
✎	00:00:39:22	10:05:21:05	Hand held camera
🕓	00:00:40:10	10:05:21:18	Cross-fade ends
✎	00:01:01:16	10:05:42:24	Sailor agains sea back ground
✎	00:01:03:15	10:05:44:23	Sea sick camera
✎	00:01:09:05	10:05:50:13	Monkey shot
🎥	00:01:13:08	10:05:54:16	Monkey looking cute
✎	00:01:18:18	10:06:00:01	Flying squirrel
🕓	00:01:22:02	10:06:03:10	Squirrel in mid flight
⚡	00:04:37:17	10:14:05:09	Timecode offset: 10:09:27:17

Time scale: 1 minute - 10 seconds

Shot comment: Flying squirrel

Timed commnet: Squirrel in mid flight

The next control simply shows the filename of the opened file. Following this is a button called Shots which starts a shot change detector running on the file. This item can also be accessed using the popup list linked to the Tools button or by typing Ctrl-Alt-S. When the shot change detector is running, an animated egg timer runs in the top panel just to the left of the file title. The application can still be used to comment the content and, if a comment is added beyond where the shot detector has reached, it will be assigned to the correct clip once the detector has completed its work.

The following three buttons allow the user to Open, Save and Close a file. They can also be accessed by typing Ctrl-G, Ctrl-S and Ctrl-C respectively or from the application's popup menu.

Following these buttons is one marked TC which stands for Timecode. This one pops up a menu that allows the user to choose whether the application displays the system timecode or the off-tape timecode. The off-tape timecode is calculated using the timecode break information from the ingest and, provided this is correct, will match the timecode from the tape. In order to arrive at a position on the tape, you must use the entire timecode break record and beginning from the starting timecode to ensure you get the correct frame. This seems a bit fussy but it is occasionally important to get the correct frame position if the material is bought in and different sections come from different owners.

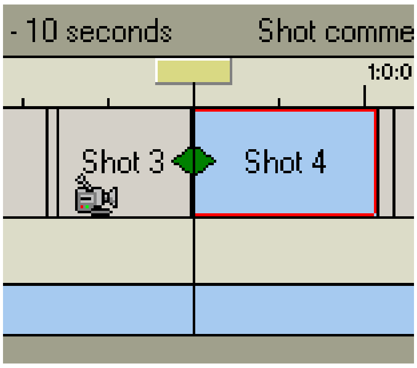
Following this is a caption which indicates which type of timecode is displayed and a timecode display. The tools menu pops up a list of property pages and tools available from the application. Last but not least is the popup menu for the application's general settings which is accessed by right clicking the top panel. This allows you to choose the colours of the application and to choose whether you want to show tool tip hints.

4.2 Control Buttons

The media player controls are standard and allow setting of the player to the start, fast reverse play, stepping backwards a frame, stopping replay, starting play, stepping forwards a frame, fast forward play and moving to the end. The fast play buttons accept multiple presses which steps the replay speed to ±12, ±24, ±48 and ±128 times normal speed.

The camera button allows you to enter a key frame and a comment at the player's current position. Note this tool does not stop replay but uses the timecode at which the control was clicked.

The three shot buttons which contain the shot icon allow insertion of a new shot change at the player position, editing of a shot's title or removal of a shot. Although these buttons can be used while the player is still playing and will use the time at which the button was pressed, it would be a bit tricky to get accurate positions so it is a good idea to step the player backwards and forwards, to choose the correct frame for the new shot, before inserting it. The insert button shows a dialog for entry of the shot name, the edit and delete buttons show dialogs that contains the existing name and invites the user to change or delete it. The application will warn the user that it will not action a change to the same name. The Delete button will ask if it should really delete the shot and provide its name.



The two buttons showing brackets allow the positions of a shot's start and stop times to be adjusted. They cause the timeline to show an adjustment Icon, shown in the graphic, over the point to be adjusted and set the player to that time position. Clicking the right or left hand arrow moves the position of the shot to a later or earlier frame and moves the position of the player to allow an accurate position to be set. It is worth saying that the shot change detector used in this application does not make many mistakes. This facility is provided for correcting shot positions from other applications. Not that in a logging file, all clips are defined as contiguous so adjusting a clip's in point also adjusts the adjacent clip's out point and vice versa.

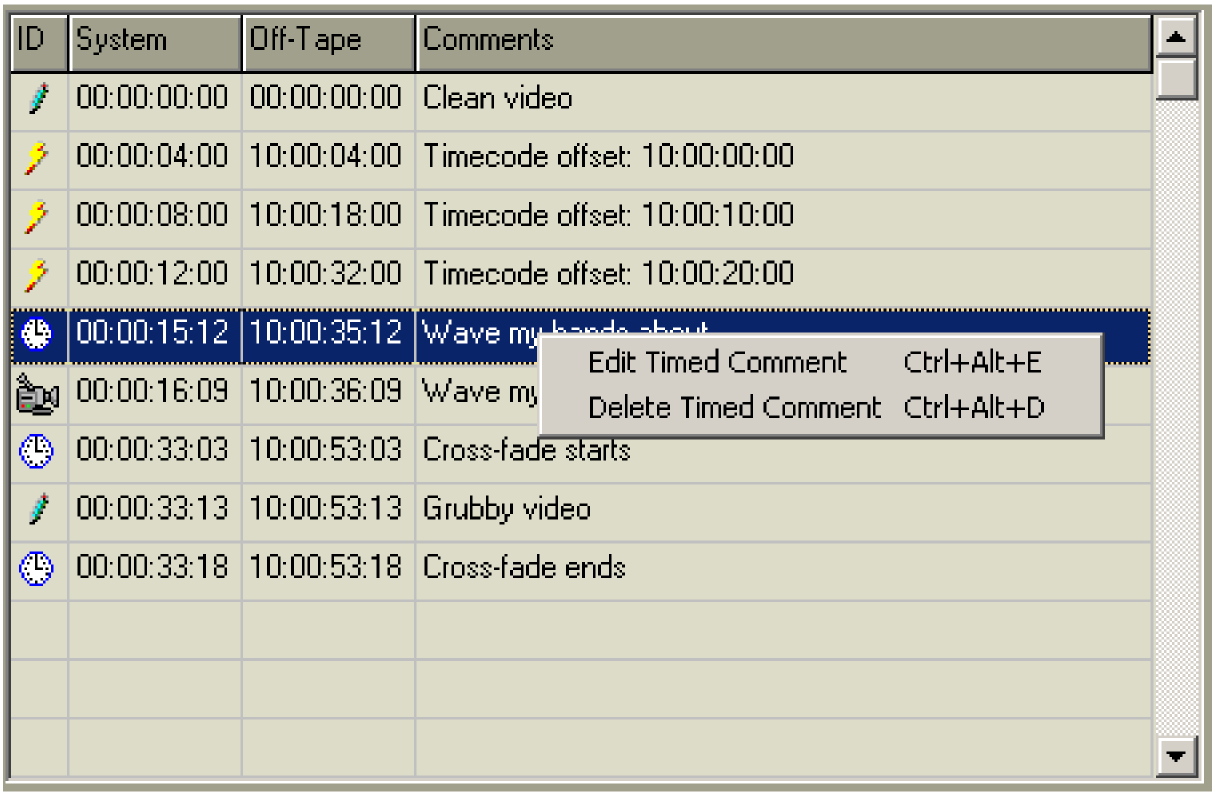
The buttons that contain a pen icon allow shot comments to be added, edited or removed. They will only be enabled if their action is valid for the currently selected clip. If there is already a shot comment there, the add button will be disabled but the edit and delete buttons will be enabled. If there is no shot there, the add button will be enabled and the edit and delete buttons will not, since there is nothing to edit or delete. The comment is attached to the shot that is currently under the timeline cursor. This shot will be selected and coloured in blue with a red border.

The three buttons with a clock in them allow addition, editing and deleting of timed comments. These comments are related to times within the selected shot and there can be more than one. If there is no timed comment in a shot the edit and delete buttons will be disabled since there is nothing to be edited or deleted. If you attempt to put a new timed comment on top of an existing one the application will tell you that it has failed to insert our comment. This avoids polling the aaf file's comment mechanisms for each new media position. If there are timed comments, the edit and delete buttons will be enabled and the application shows a dialog containing the title of the comment to be edited or deleted. This will be the nearest comment to the player's current position.

All of these controls do not stop the player, since a really slick operation may be able to comment a file dynamically as the play progresses.

4.3 The comments table

This contains a list of comments ordered according to their system timecode. A second column shows the off-tape timecode. The type of comment is indicated by the icons in the first column. This example shows a list of timecode breaks at the start of the media, a Key Frame and some shot and timed comments.



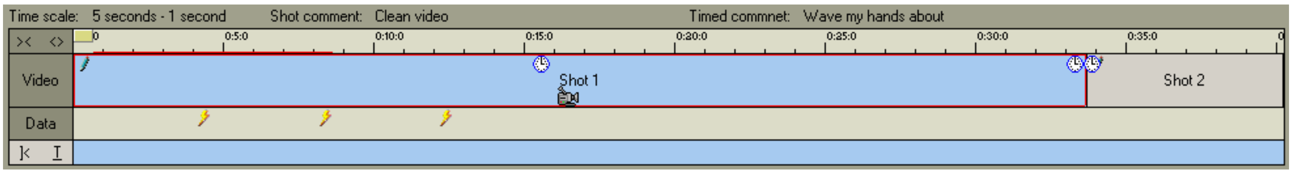
ID	System	Off-Tape	Comments
✎	00:00:00:00	00:00:00:00	Clean video
⚡	00:00:04:00	10:00:04:00	Timecode offset: 10:00:00:00
⚡	00:00:08:00	10:00:18:00	Timecode offset: 10:00:10:00
⚡	00:00:12:00	10:00:32:00	Timecode offset: 10:00:20:00
🕒	00:00:15:12	10:00:35:12	Wave my hands about
🎥	00:00:16:09	10:00:36:09	Wave my
🕒	00:00:33:03	10:00:53:03	Cross-fade starts
✎	00:00:33:13	10:00:53:13	Grubby video
🕒	00:00:33:18	10:00:53:18	Cross-fade ends

Edit Timed Comment Ctrl+Alt+E
Delete Timed Comment Ctrl+Alt+D

Left clicking a row sets the player position to the position of the comment. Right clicking the table shows a dialog that enables you to Edit or Delete the selected comment. Note the selected comment is highlighted but if the highlighted row is empty the dialog will read 'No Popup Items Available'.

4.4 The timeline

The timeline shows the video shots and non clip related data, which in this application is just the timecode breaks. The video clips show Icons for key frames, shot and timed comments. Clicking an icon sets the player position to that of the comment and selects it in the comments table to allow checking of the content and editing or deleting the comment.



Time scale: 5 seconds - 1 second Shot comment: Clean video Timed commnet: Wave my hands about

The timeline also displays the current timescale for the major and minor timeline ticks, and the values of the shot comment and nearest timed comment to the cursor – if there are any.

The timeline ruler also shows the timescale in hours, minutes, seconds and frames with suppression or the hours if they are zero. At the top left of the timeline are two buttons that allow stepping of the timescale through a range of preset values, which changes the timeline ruler and clip displays.

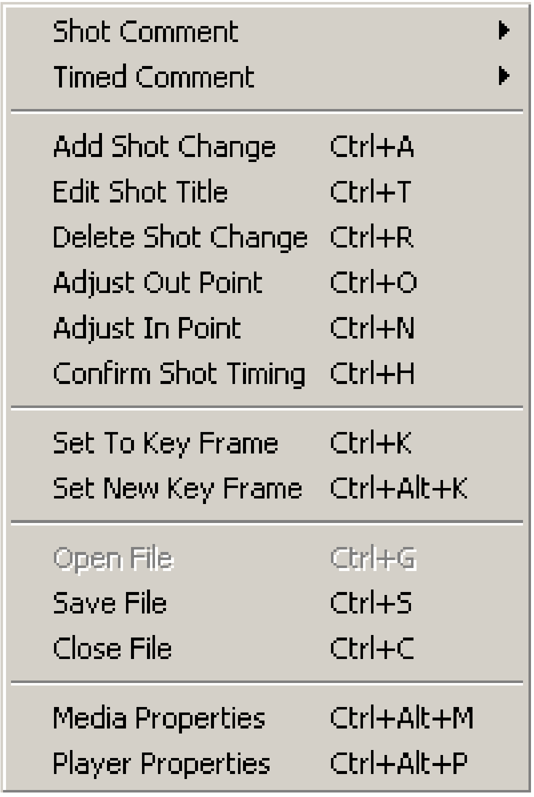
The timeline ruler can also be adjusted by dragging the mouse on it. If the Control Key is down, the ruler position will follow the mouse to allow you to move along the timeline. If the Alt key is down then the timeline scale is varied smoothly, with the cursor position remaining unchanged. This allows zooming in on the cursor position for a closer look.

Dragging on the ruler with no buttons pressed allows selection of start and stop times which will be displayed by a red bar spanning the mouse down and mouse up points. This will enable the Play Selection button which can then be used to play the selection. A single click on the ruler removes the selection bar.

The ruler position can be reset to the start of the media by clicking the |< button at the lower left. If the Control key is down, the timeline scale will be adjusted so that the media fills the timeline. The remaining button at the lower left sets the timeline scale to one of a pre-determined set.

The player position can be dragged by dragging the timeline cursor or set to the start of a clip by clicking the clip on the timeline.

4.5 The Application Popup Menu

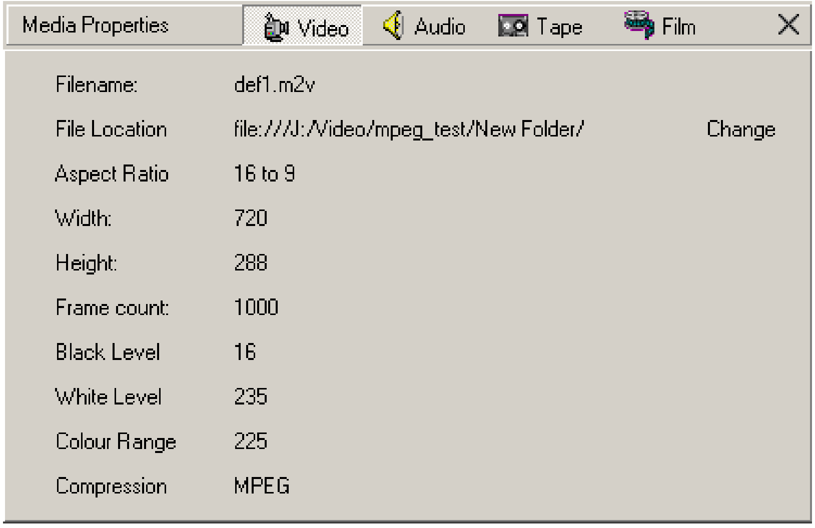


Right clicking the timeline or the media player pops up the Application Popup Menu. This contains a list of most of the actions a user may want to do. It has two sub menus, one for adding, editing and deleting shot comments and the other for doing the same for timed comments. The sub-menu items are enabled according to whether there are any valid actions allowed.

The next section contains actions for processing the shots in the timeline. These are adding a shot, editing its title and removing it. The action is carried out at the player's current position. Again, an attempt to insert a new shot at the starting position of an existing shot will fail. There are also items for adjusting the in or out points of the selected shot which show the adjustment icon on the timeline. An adjustment can be confirmed using the Confirm Shot Timing Icon, which removes the adjustment Icon. You cannot move a shot change outside the range of its preceding or succeeding shot.

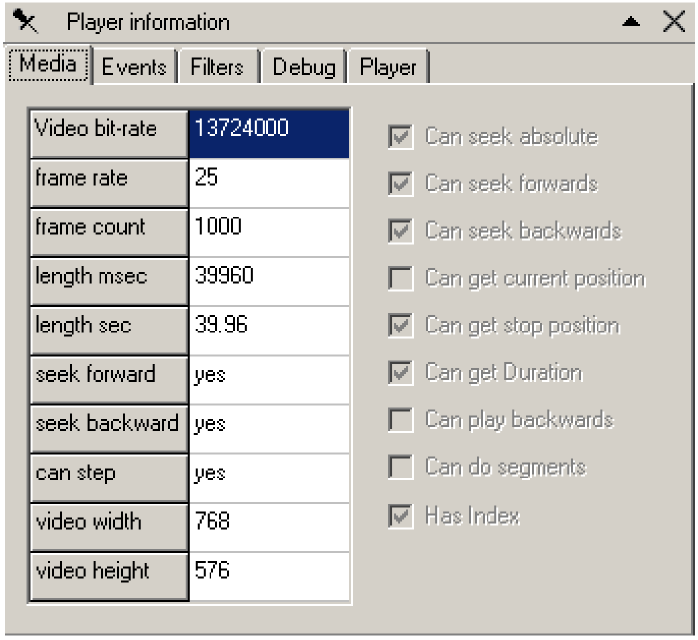
Open, Save and Close are self explanatory.

Media and Player Properties display dialogs that shows the properties of the media and the player.



Filename:	def1.m2v
File Location	file:///J:/Video/mpeg_test/New Folder/
Aspect Ratio	16 to 9
Width:	720
Height:	288
Frame count:	1000
Black Level	16
White Level	235
Colour Range	225
Compression	MPEG

The Media Properties dialog shows a range of stored properties pulled from the file. This has several tabs selected by buttons along the top. Some of the properties have been made editable to show that it is possible. I chose the locations of the video and audio media files and the tape and film since these are things that users may well want to change. The dialog does not show import or recording information yet.



Video bit-rate	13724000
frame rate	25
frame count	1000
length msec	39960
length sec	39.96
seek forward	yes
seek backward	yes
can step	yes
video width	768
video height	576

☑ Can seek absolute

☑ Can seek forwards

☑ Can seek backwards

□ Can get current position

☑ Can get stop position

☑ Can get Duration

□ Can play backwards

□ Can do segments

☑ Has Index

The data held in the Tape information is Name, Location, Form Factor, Tape Format, Manufacturer, Vtr Type, Video Type, Length, Audio count and whether the material has timecode breaks. The data held for the film is Name, Location. Format, Frame Rate, Aspect Ratio, Manufacturer, Type and Audio Count.

The Player properties shows data about the video file including its media length and whether the player believes it can seek through the file. The dialog also shows DirectShow events and debugging information which are not much use to most users, The filters tab shows lists of the DirectShow filters installed on the machine and the Player Tab allows the location of the player's Temp folder to be changed.

4.6 The Tools menu

The tools menu provides access to the tools available in the application. The first of these is a diagnostic called MemStatus which shows the memory usage of the application. The method used to access the aaf sdk involves a great deal of class construction and destruction so this helps ensure that the code does not forget to destroy any classes it creates. (The aaf wrapper code also includes an option to log the creation and destruction of classes and lists the remaining classes when the application is closed).

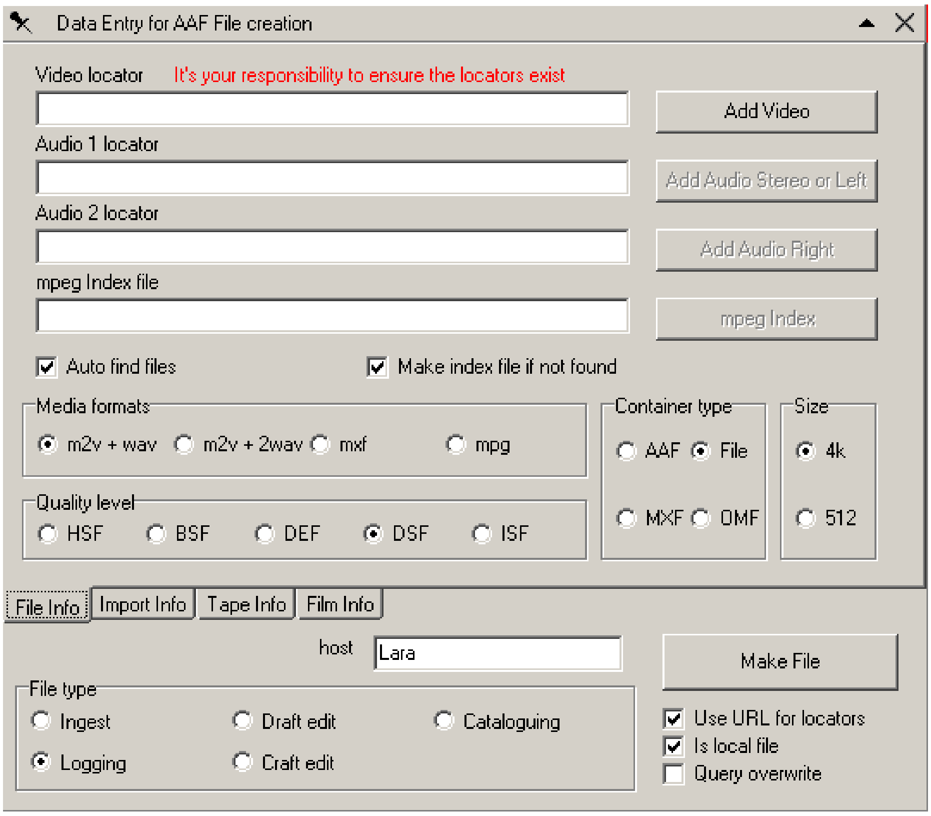
The second tool is Make AAF File. This one can be used where the media is delivered as media files rather than pointed to by an aaf file. It launches a dialog that allows the parameters of an aaf file to be set. This process if fairly complicated and is described in the following section.

The third tool is Shot Change which simply runs a shot change detector on the media in the opened aaf file.

The final tool is Correct Shot Names. This tool runs through the media and sets the shot names to Shot x where x starts at 1 and increases through the media. This allows the clips to be given names that are unique within this file, if a little boring. It should not be use if the file has had more imaginative names assigned to the clips since it will over write them.

5 Making a new aaf file

This part of the application allows a user to make a new aaf file when no Ingest or Media Asset Manager has delivered one. It should not be required in a fully fledged media system. The process is broken into four parts.


Data Entry for AAF File creation

Video locator It's your responsibility to ensure the locators exist

Audio 1 locator

Audio 2 locator

mpeg Index file

☑ Auto find files ☑ Make index file if not found

Media formats: ● m2v + wav ○ m2v + 2wav ○ mxf ○ mpg

Container type: ○ AAF ● File ○ MXF ○ OMF

Size: ● 4k ○ 512

Quality level: ○ HSF ○ BSF ○ DEF ● DSF ○ ISF

host Lara

File type: ○ Ingest ○ Draft edit ○ Cataloguing ● Logging ○ Craft edit

☑ Use URL for locators
☑ Is local file
☐ Query overwrite

The first operation is to name the Media Files and you start by naming the video file.

The check box called Auto find files attempts to find matching audio filenames. These have the same file stem as the video but have a wav extension. There can be one or two files depending upon what Media Format is selected.

The Quality level parameter is intended to mark the media as High Definition Stored Format (HSF), Broadcast Stored Format (BSF), Desktop Editing Format (DEF), Desktop Stored Format (DSF) and Internet Stored Format.

These types of format are linked to the capabilities of Professional editing applications which do not focus on handling of mpeg files. HSF and BSF are full quality formats and in this application refer to Mpeg High Level and Main Level 422 formats. DEF reflects the preference of professional editors to use formats that are not temporally compressed and in this application refers to Mpeg I-Frame only format. DSF is intended to achieve a good spatial resolution at the minimal bit-rate and in this application refers to Mpeg Main Profile using temporal compression and variable bit rate. ISF is a format for internet access and could for instance be Mpeg 1.

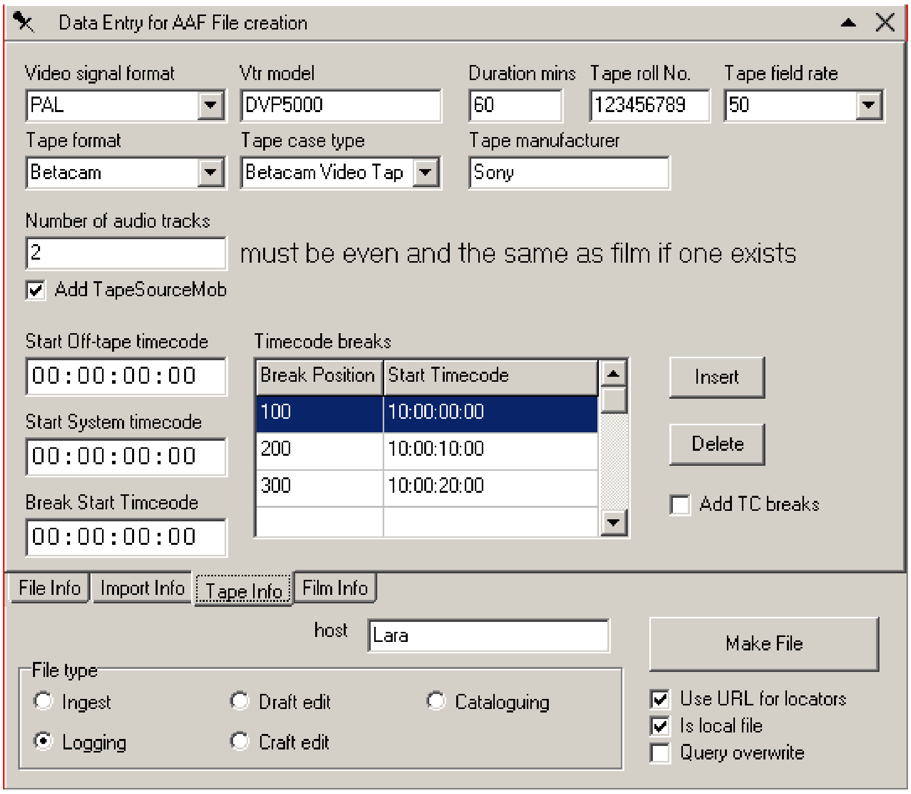
These formats are examples only, although they are sufficiently mature that there are already applications that can offer broadcast levels of usability. This may not be the case for more modern compression formats. The purpose of having this parameter is to model ingest systems that capture more than none media format at the same time, which avoids having to do manual conversions and ensures only one media duration for all instances.

The application also allows the container type to be set. Only File is implemented currently. The file can also have its Structured Storage type set and the default is 4 kilobytes.

Of the file types, only Ingest and Logging are currently implemented and the application expects a logging file. An ingest file contains only the media and might be what an ingest system would deliver to a MAM.

The make file button shows a dialog that allows the user to choose the location and filename of the created aaf file. The three check boxes below this button control whether the application inserts a simple filename or a URL for the Network Locator which points to the media, whether the file is local or remote and whether to query the overwriting of an existing file of the same name.

The Import Info Tab allows entry of either an Import or a Recording Descriptor. An Import Descriptor is used when media is imported into an application from a remote on-line source. A file with an Import Descriptor can still have a Tape and/or a Film derivation chain. A Recording Descriptor, on the other hand, is used where there is not physical entity to be derived from. If a file has a Recording Descriptor there should be no Tape or File derivation chain.



Data Entry for AAF File creation

Video signal format: PAL
Vtr model: DVP5000
Duration mins: 60
Tape roll No.: 123456789
Tape field rate: 50
Tape format: Betacam
Tape case type: Betacam Video Tap
Tape manufacturer: Sony
Number of audio tracks: 2 must be even and the same as film if one exists
☑ Add TapeSourceMob
Start Off-tape timecode: 00:00:00:00
Start System timecode: 00:00:00:00
Break Start Timceode: 00:00:00:00

Timecode breaks

Break Position	Start Timecode
100	10:00:00:00
200	10:00:10:00
300	10:00:20:00

☐ Add TC breaks
host: Lara
File type: ○ Ingest ○ Draft edit ○ Cataloguing ● Logging ○ Craft edit
☑ Use URL for locators
☑ Is local file
☐ Query overwrite

The Tape Info tab allows entry of information about the video tape from which the media was taken. Most of the data on this tab is self explanatory. Where the values are constrained, entry is from a drop-down list.

The Tab also contains controls for simulating a timecode break list. The Start Off-Tape timecode box allows entry of the starting timecode from the tape and the Start System timecode defines the system timecode at which the replay started since this may not have been zero.

Timecode breaks are added by first setting the starting time of the off tape timecode into the Break Start Timecode control, then clicking the Insert button. This will show a dialog that asks for the system frame count at which the break happened. Once this has been entered the timecode break appears in the grid. The grid is not ordered so you will need to position the selected row accordingly. To add the list of timecode breaks to the file click the Add TC Breaks control before creating the file.

The Film Info tab contains fields for entering the details of the film and where the parameters are constrained, entry uses a drop down list.

Both the Tape and the Film tabs have check boxes that control whether the data will be added to the file.

6 Locations of data in the aaf file

The locations of data within the aaf file conforms to the aaf Edit Protocol. The application is intended to break up an already existing composition, from an ingest or a finished programme, into its component shots. Consequently, each clip in the composition links to a separate Master Mob.

The start times of the Source Clips are zero and the start time of the clip is stored in the Master Mob Source Clips. This differs from the approach normally used in editing files where the start time of a shot is stored in the Source Clip and the source Master Mob start time is zero. This arrangement should be understandable by a fully Edit Protocol conformant editor which should sum the start times of all items between the composition source Clip and the File Mob, to find the start time. The file also contains a Master Mob that represents the media as it exists in the File.

Comments for a shot are attached to the Master Mob for the shot as either a comment on the master mob for a shot comment or a sequence of comments under an Event Mob Slot for timed comments. The Master Mob also contains an Essence Group that allows storage of the Key Frame position and caption alongside the video Source Clip.

Timecode breaks are stored in timecode objects in the Tape Master Mob. Each timecode is stored in a Sequence under its own Timeline Mob Slot and there is one timecode object in the off-tape Sequence for each timecode break. There will normally be only one timecode object for the system timecode describing the length, start time and frames per second value. For the timecode breaks there will be at least one item that describes the start time of the off-tape timecode. The item also holds the length of the timecode up to the next timecode break. If there is no further timecode break there will only be one item with a length equal to that of the system timecode object. It there are further timecode breaks then each one will have a timecode object which describes the off-tape timecode following the break and the time for which it lasts. The position on the tape can then be found by starting at the first timecode object and iterating through the list until the elapsed time is equal to the system timecode.

This metadata arrangement means that an application that breaks up the media into separate files can simply copy the Master Mob for each shot from the logging file. This will copy the comments and key frames for the shot. It can also use the powerful aaf function ‘CloneExternal’ to copy the derivation Mobs, including the Tape Mob with its list of timecode breaks, so that each shot contains a full derivation history.

7 Conclusion

This work has shown that the Edit Protocol is capable of delivering work to a logging application, documenting the metadata created during the logging, and returning the completed log to a Media Asset Manager. AAF has shown its capabilities for editing through its use in professional products. The documentation of the Edit Protocol and the availability of the aaf SDK means that system integration from ingest through to and including craft editing is greatly simplified. The proposal for joining up broadcast production outlined in this document uses simple email attachments and does not, therefore, require additional staff training.