

Multimedia In SAS® Applications

Ken Ellis, SAS Institute Inc., Cary, NC

ABSTRACT

This paper describes the multimedia offerings that will be available with Release 6.11 of the SAS System. The first section of the paper describes the SAS/AF® Video Player Class. The Video Player Class allows users to embed digital video and audio directly within FRAME based applications. A detailed first look at the Video Player Class methods is included in this section. The next section of the paper describes the Video Editor found in the SAS/GRAPH® Video Extensions. The Video Editor allows users to edit multiple video and audio clips in a non-linear domain.

A basic understanding of SAS/AF and Screen Control Language (SCL) is assumed in the programming examples. The reader should also be familiar with the basic concepts of digital video and audio. However, the paper introduces key topics including the supported video formats, non-linear editing, and compositing.

INTRODUCTION

In only a few short years multimedia has evolved from a curiosity into a powerful and rapidly expanding technology. Digital video, audio and imagery have found their way into all aspects of the scientific and corporate community. New and better tools that allow audiences to visualize complex systems that were once too difficult or cumbersome to express with traditional text and charts are constantly being developed.

Currently, multimedia is being used in such diverse fields as scientific visualization; computer based training; documentation and help facilities; and marketing presentation systems. This list is by no means complete though, as new applications of the technology are being developed daily.

Two new offerings for release 6.11, the SAS/AF Video Player Class, and the Video Editor found in the SAS/GRAPH Video Extensions, bring exciting new multimedia capabilities to the SAS System. A third tool, a video player included with base SAS software, allows users to examine first hand a playback application written using the Video Player Class.

THE VIDEO PLAYER CLASS

The Video Player Class allows application developers to embed digital video directly into FRAME based applications. Video can be combined with other media types such as text and static images to create powerful presentations.

A variety of video formats are supported across all 6.11 platforms including Microsoft's popular AVI (Audio Video Interleave) and MPEG-1 (Motion Picture Experts Group). The device independent design of the Video Player insures developers that all application will have the same look and feel on all supported platforms. (For more information on video file types, see SUPPORTED VIDEO FILE FORMATS.)

Video Player Class Methods

The method set supplied for the Video Player Class allows precise control over most aspects of video and audio playback. Some of the more commonly used methods include:

MUTE

Mutes the audio output.

PLAY

Plays the video starting with the current frame.

_PLAY_SEGMENT_

Plays a selected segment of the video.

_SCALE_TO_REGION_

Scales the video to fit the playback region while maintaining the original aspect ratio of the video.

_SET_AUDIO_BALANCE_

Sets the ratio of left to right stereo audio output.

_SET_AUDIO_DESTINATION_

Sets the output destination to an internal or external system.

_SET_AUDIO_STATE_

Enables or disables audio playback.

_SET_AUDIO_VOLUME_

Sets the audio volume level.

_SET_FRAME_NUMBER_

Sets the current frame to a user supplied number.

_SET_FRAME_SCALE_

Scales the original video size by a factor.

_SET_PLAY_MODE_

Sets the playback mode to single or continuous play.

_SET_PLAY_SPEED_

Sets the playback speed as a ratio of the video's default playback speed.

_SET_TEXT_

Specifies the filename of the video to be played.

_SET_VIDEO_STATE_

Enables or disables video playback updates.

_SHOW_FRAME_

Shows the current video frame.

STOP

Stops video playback.

Example

The following code segment demonstrates the use of the Video Player Class.

First, we invoke the SAS/AF BUILD procedure and create a FRAME entry with a video region and 2 push-button widgets. The video region is named *video* and the two push-buttons are named *play* and *stop* respectively.

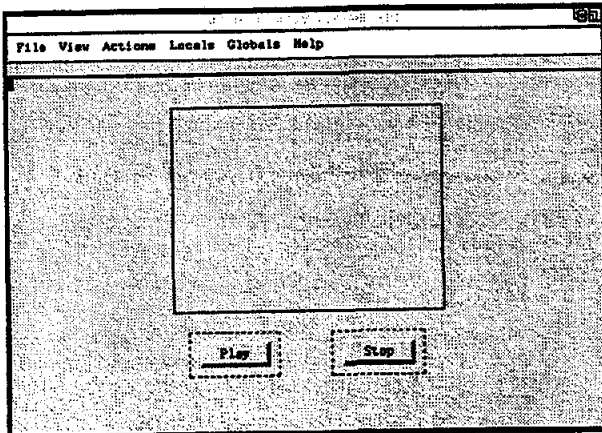


Figure 1 FRAME entry with push-buttons

Next, we create the SCL code that will drive the FRAME application. In the *init* section we open the video and setup our initial playback characteristics. The *play* and *stop* sections are coded to call the corresponding Video Object methods. Finally, we compile the SCL code and test the application.

```

BUILD: SOURCE A.SCL (E)
File Edit View Locals Globals Help

00001 /* Copyright(c) 1995 by SAS Institute Inc., Cary, NC USA */
00002
00003 init:
00004   call notify("video", "_set_text_", "/video/right.avi");
00005   call notify("video", "_set_frame_number_", 1);
00006   call notify("video", "_set_frame_scale_", 2.0);
00007   call notify("video", "_set_audio_volume_", 0.0);
00008   call notify("video", "_show_frame_");
00009 return;
00010
00011 play:
00012   call notify("video", "_play_");
00013 return;
00014
00015 stop:
00016   call notify("video", "_stop_");
00017 return;
00018

```

Figure 2 SCL Code

SUPPORTED VIDEO FILE FORMATS

The Video Player Class supports playback of AVI, and MPEG-1 encoded files. These two formats, the former a software-based solution initially developed for personal computers, and the latter, a typically hardware assisted up-and-comer, comprise the lion's share of the digital video market today. Each format has its own relative strengths and weaknesses.

The AVI format

The AVI format developed by the Microsoft Corporation has two major strengths, or weaknesses, depending on your viewpoint. A brief discussion of each follows.

First, the AVI solution is totally software based and does not require any specialized hardware assistance to aid in video or audio stream playback. The host's CPU processes the video and audio streams on-the-fly. While this scheme works well for CPUs with the necessary horsepower, it can quickly overwhelm less powerful CPUs, leaving video playback looking more like a slide show with sound!

The second major feature of the AVI format is its extensibility. The AVI file specification does not recommend any certain video or audio compression or decompression algorithms (codecs). This means that third party vendors are free supply custom codecs that users can install on their systems. This virtually guarantees that a state of the art codec can, and will be, supplied to the AVI community. At first glance, this would seem to be an advantage for the AVI format, but remember, to play an encoded AVI file you must first have the encoding codec installed on your system.

The MPEG Format

The MPEG standard defines an architecture that supports a wide range of video resolutions and data rates. Of particular interest to computer based video users is the MPEG-1 standard that is optimized for data rates typically used by CD-ROM drives (about 1.2Mbps) at a resolution of 352 x 240 pixels.

As was mentioned earlier, MPEG-1 encoded files typically require hardware assistance to facilitate playback. This is due to the complex algorithm used during the compression process. Once again, depending on your point of view, this could be a plus or a minus. On the plus side, hardware assisted playback relieves the host CPU from the burden of decompressing video and audio streams. Also, this scheme makes video playback more accessible to low end systems via add-on boards equipped with MPEG decoding chipsets. The downside of this scheme is that computer owners have to buy specialized hardware to play MPEG encoded files. This may not be the case in the near future though, as fast software MPEG-1 codecs are being developed, and should be available soon.

THE VIDEO EDITOR

Introduction

The previous sections of this paper have focused primarily on video playback. Eventually, you may want to create customized video to meet the needs of a specific application or project. The Video Editor supplied with the SAS/GRAPH Video Extensions, is a powerful tool that helps you do just that. The Video Editor combines the power of a non-linear editing tool and a compositor in one easy-to-use package.

The Video Editor supports the popular AVI video and WAV (Microsoft Waveform) audio file formats for input. MPEG-1 video files are not supported as an input format. This is primarily due to the high cost of software decompression associated with MPEG-1 encoded files. All output is written to the AVI format.

Historical Perspective

Traditionally, video tape or film editing has been a very time consuming and error prone process. For example, to insert a film segment in the middle of another film segment, pieces of tape had to be physically cut and pasted. Once a cut was made, you were stuck with the result.

This is not the case with digital non-linear editors. Non-linear editors work with digitized video clips so there is no physical tape involved in the editing process. Video segments can be cut, pasted, inserted, and reordered countless times without any loss of image or audio quality. The digital video is transferred to tape only when you are satisfied with the final production.

Starting the Video Editor

Invoke the VEDIT procedure from the command line to start the Video Editor. Once started, the Construction and Project windows are displayed. These two windows provide an interface from which users can edit video, create transitions between video segments, and overlay audio to form finished video clips.

The Project Window

The Project window is used to group video and audio clips that will be used in an editing session. To include a file in the current group, simply click the left mouse button on the File menu and select the Open option. You will then be prompted, via a standard file selection box, to choose the file that you wish to load into the Project window group. Once loaded, a thumbnail image of the file, and the file's vital statistics, are displayed. You can then preview or set *in* and *out* points (starting and ending frame ranges) before the file is used in the edit session.

The Construction Window

The Construction window is the heart of the Video Editor. This is where the actual video and audio editing is done. Clips are transferred to the editor's tracks from the Project window and are then combined via transitions to form a final video.

Example

For this example two videos are combined to create a finished video clip. We will use the first ten seconds of GEESE.AVI and the first five seconds of GIRL.AVI to create the fifteen second finished video. We use a two second *dissolve* transition between the two clips.

First, we import the two video files into the Project window.

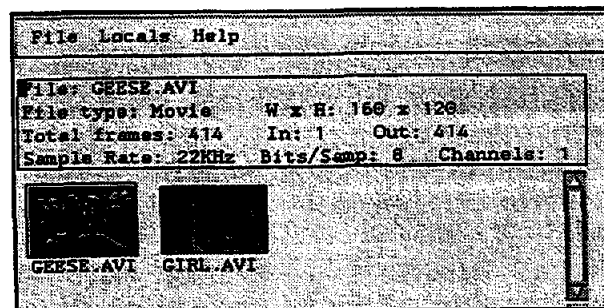


Figure 3 The Project window with two video clips

Next we *drag-and-drop* GEESE.AVI and GIRL.AVI from the Project window into the Construction window's A and B Video tracks respectively. GEESE.AVI is aligned at time reference 0.00 seconds and GIRL.AVI is aligned at 10:00 seconds.

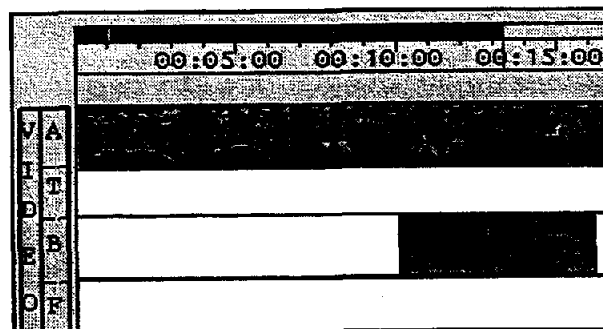


Figure 4 The Construction window Video track setup

We now drag-and-drop the dissolve transition into the Construction window's Transition track at time reference 10.00. Finally, we use the Construction window's *resize* tool to set the dissolve time to two seconds.

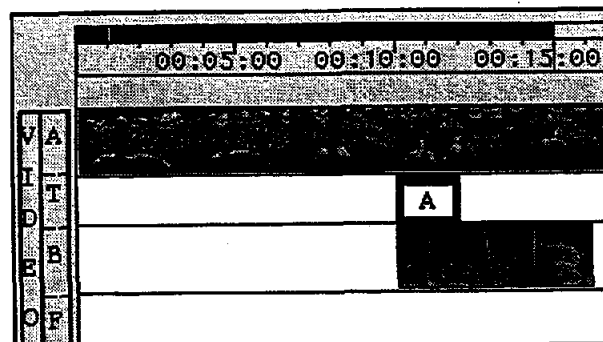


Figure 5 Adding the Dissolve transition

At this point we can either watch a preview of our video or save our finished video to a file. To preview, select the **Rebuild** button to build a temporary copy of the finished video. Then select the **Preview** option from the **Locals** menu to play the preview. To write out the finished video, select the **Save as...** option from the **File** menu. You will be prompted for a filename for the output file.

The Compositor Window

The Video Editor's Compositor window provides you with a set of tools that help you create and overlay professional looking text onto your video. For example, you can create "rolling credits" that you typically see at the end of movies, or text that fades in and out at predetermined frames of your video.

The key is the Compositor window's built-in animation system. The animator gives you direct control of text position and visibility over any range of a video clip's frames. You simply enter the position and visibility of the text at certain *key* frames of the video and the animation system interpolates the text's movement and visibility between each key. Once you are satisfied with your animation, you can render the entire sequence and then import it back into one of the Construction window's Video tracks.

The compositor supports a large number of user selectable font styles. Text attributes such as size, spacing, justification, and color can also be modified.

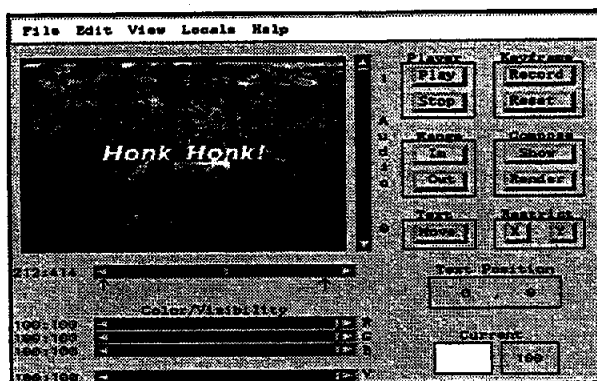


Figure 6 The Compositor window

CONCLUSION

Each day we are barraged with new and more complex forms of data. Sifting through and trying to make sense of it all is a very time consuming process. New methods for communicating and visualizing increasing mountains of data must be developed.

The new multimedia tools supplied with Release 6.11 of the SAS System have been developed to give application developers and users state-of-the-art methods to create and experience data.

The SAS/AF Video Player Class gives developers a simple, yet powerful, means to embed digital video and audio directly into SAS applications in a completely portable manner.

The Video Editor, shipped with the SAS/GRAPH Video Extensions, allows all SAS system users to produce quality video presentations.

ACKNOWLEDGEMENTS

SAS, SAS/AF, and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.