Developer Support Application Note
Number: 206
Issue: 1
Author: R Wilson

Developers' Notes: Acorn Replay Implementation Details

This application note describes the programmer's interface to the Acorn Replay decompression code. It is necessary reading for any developers intending to embed Acorn Replay support into their code.

Applicable Hardware : A4XX
A3000
A5000

Related Application Notes: Acorn Replay - Rules for Use

Support Group
Acorn Computers Limited
Acorn House
Vision Park
Histon
Cambridge CB4 4AE

Introduction

The software for replaying movies resides in the directory !ARMovie (think of it like a rather large system directory - currently about 640KBytes). The !Boot file for !ARMovie sets up the computer to play movies:

- loads the iconsprites for the AE7 filetype
- sets File$Type for AE7
- sets the RunType alias for AE7 to point to the Player program
- sets ARMovie$Dir so that the Player program knows where it is (there are dither tables and sound code in the directory)
- ensures the computer has ColourTrans loaded (from the system directory - any version will do)

After this, double clicking on an Acorn Replay Movie will play the movie on the desktop. Just running the program at the cli will play it. A program can do:

```
SYS"Wimp_StartTask","movie file name"
```

to play the movie.

The program attached to !ARMovie is a minimal desktop viewer for replay files - it can display the helpful sprite and allows desktop control of a few of the capabilities of the Acorn Replay Player program.

Note that the Player program has to take over the entire machine in order to get its work done. It consists of independent interrupt run processes:

- file chunk loader
- master time 'conductor'
- video decompressor
- video frame refresh and dither
- sound player
- mouse event handler

The video sections of this code are custom assembled to suit the particular screen mode etc.

The interrupt run processes are communicating using shared memory in the ordinary virtual address space - a wimp task swap would be fatal. A large amount of memory is required:

- double buffer for the file I/O (odd and even chunk size given in the AE7 file header)
- double buffered queue of decompressed frames
- queue of sound chunks

The amount of memory required is at least:

"odd" chunk size + "even" chunk size + 320K
+ 48K (or more) if sound is to be played
+ 32K to 128K for dithering tables

The Player program can use more memory if it is available! It will (by default) return with a textual error if there isn't enough memory.

A major limitation of the video frame refresh and dither section of the program is that it uses word addressing in order to paint on the screen: this results in the displayed section be somewhat to the left or right of the desired position. With 16bpp it can be 1 pixel out, 8 bpp 4 pixels, 4bpp 8 pixels, 2bpp 16 pixels, 1bpp 8 pixels. The program rounds the pixel address to the word address.

Simple Control

The Player program has some controls hard wired into it:

Esc

The Esc key terminates a display sequence.

Menu (Mouse middle button)

The Menu button (also) terminates a display sequence.

Adjust (Mouse right button)

The Adjust button pauses video and sound while it is held down.

Other capabilities may be provided as icons by the calling program.

Parameters

The Player program also accepts a list of parameters on the command line after the Acorn Replay movie file. These are

-At <Number>,<Number>

Specifies the position of the bottom left corner of the replayed movie in OS units x,y. Player will move the output display such that it is entirely on screen. Note the comments above about precision of positioning the output.

-Big

If the movie is <=160 X, <=128 Y, then Player will play it on a Mode 13 ie. 320 by 256 pixel screen filling the screen. If the movie is <=192 X, <=144 Y then player will install a temporary mode 96 which gives an overscanned Mode 13 display and play the movie filling the screen.

-Leave

By default, Player removes the output from the display. If -Leave is specified it will leave the output on the display when the program is left (either by getting to the end or Escape/ Menu). Leave state is forced if the user exits by clicking on the CAPTURE/EXIT button.

-Loop

Player will loop back to the start of the movie when it gets to the end, thus playing for ever. This overrides -playfor.

-Mute

Player starts up with sound initially turned off.

-NoAdjust

Suppresses the use of the Adjust mouse button for Replay control.

-NoError

Suppresses all possible textual errors from the Player program. The variable ARMovie$ReturnCode will contain the error.

-NoMenu

Suppresses the use of the Menu mouse button for Replay control.

-PlayFor <Number>

<Number> represents a time in centiseconds: this amount of the movie is played and the program exits. Note that Pause, fast forward, half speed etc. do not affect this time limit - it represents a duration in frames (or feet of the original film) rather than a time in the future.

-Quiet

Disables the sound permanently. (Unlike -Mute it cannot be turned back on with the controls)

-Relative

Changes button definitions such that the origin is relative to the -at value rather than absolute.

-Startat <Number>

<Number> represents a time in centiseconds: the movie is played from this point. It can be very time consuming to reach a random point in a none key frame movie, but Player can do it if necessary. Key frame movies should only take a few seconds to start playing at any point.

-Silent

Starts playing with sound initially disabled.

In addition to those parameters, the calling program can specify the positions of control buttons on screen. The Player program caters for buttons with the following functions:

exit
capture/exit (exactly like exit but leaves pixel map on screen)
pause
single frame advance when in pause mode
fast forward
half speed
quiet (no sound)
loud (sound back on)
mute (toggle sound)

By default no button definitions exist except for pause, defined as the whole screen: so by default a left click will pause the movie and another will restart it. Buttons are defined with the initial character followed by the bottom left corner i.e. E10,40 - this gives a default sized button of 32 OS units. E10,40,50 gives a square button of 50 OS units. E10,40,50,25 gives a rectangular button 50 OS units wide, 25 tall. (-relative makes button definitions relative to the -at position). Defining any button position will remove the effect of the default pause button; use -NoMenu and -NoAdjust if you need to prevent the other mouse button controls.

A typical command line might be:

<movie name> -loop -leave -startat 2000 -playfor 2000 -quiet E640,512 P0,0

case is irrelevant.

System Variables

There are several system variables which affect the Acorn Replay player program:

ARMovie$Place

Contains the position on screen where the player will put the movie as <leftx> <bottomy> - for example *set ARMovie$Place 640 512

Controlling programs must set this immediately before running the movie.

Value is only used if -At parameter is not given.

It is recommended that -At be used in preference to this variable.

ARMovie$PrefMode

Used by the machine's owner to get the movie to display in a different mode to the current one. For example, a user may like working in a very high resolution mode with a low number of colours. Player can display movies in such modes but they don't look good, so the user can set ARMovie$PrefMode to an alternative mode. Note that the user won't be able to see the buttons. (Currently ARMovie$PrefMode is the only way to get playback in 16 bit per pixel modes, since the desktop doesn't operate in those modes).

If ARMovie$PrefMode is 13, then -Big is assumed.

[Aside: minimal mode if -1 . Player installs a 320x128 pixel temporary Mode 96 with the same timings as Mode 15.]

ARMovie$Suffix

Used by the machine's owner to preselect different versions of the movie if they are available. For example, if one knows that one's machine can only manage 12.5fps movies, *set ARMovie$Suffix to 2 and Player will preferentially play the 12.5fps movies. See the file on AE7 file naming conventions.

ARMovie$Col

New variable being implemented - specification to follow

Some system variables are used by Player to return information to the calling programs:

ARMovie$Return

Player sets this to the actual position on screen of the video information. If -leave has been set or user has exited Player using CAPTURE/EXIT button then this is the section that has been 'dirtied' and should be repainted by the window manager. It is set to

<leftx> <bottomy> <rightx> <topy> <C>

Where the <C> is present only if the user has pressed CAPTURE/EXIT.

ARMovie$ReturnCode

If -NoError has been set, this will contain the textual error message. It will be null if there was no error.

ARMovie$Sound

Player sets this to 0 if the sound has been disabled by clicking on one of the icons, 1 if it is enabled. This, plus the use of -mute, allows sound on/off state to be preserved between the applications and the Player program.

ARMovie$Time

Player sets this to the number of centiseconds of movie which have been played. Note that (like -playfor) this refers strictly to the amount of the movie, not the total elapsed time: playing a movie and playing the movie pausing every frame for an hour will end up with the same value for ARMovie$Time.

ARMovie$Version

Player sets this to its version number and date string.

The !ARMovie application has already *iconsprite'd the icons for the AE7 filetype into the window manager's icon pool - you should not include them (or any other system icons...) in your application's own !Sprite files.

A standard set of icons is made available for the buttons used to control the player. They are in <ARMovie$Dir>.Sprites (plus Sprites22 and Sprites23 using the RISC OS 3.00 discipline for alternative resolutions). This gives all eight buttons with names 'exit' 'pause' 'single' 'fast' 'half' 'quiet' 'loud' and 'mute'. In addition, there are 'play' and 'pplay' (pushed play) buttons to complete a control panel.

Applications wishing to use the button set should load (check for failure) the sprite file into their own sprite area (rather than *iconsprite'ing it - the definitions use up quite a bit of memory).

If you wish to use your own icons for control buttons (for example, you want the buttons to be a different size) then please ensure that the buttons appearance (especially the shape of the glyph design) is close to the standard ones.

The sprite file <ARMovie$Dir>.Default contains the default sprite used by the !ARMovie program to indicate that an ARMovie file can be dragged there.

Note: to be implemented

The player does not, in fact, respond to double speed and half speed play.