MP_STE v3.04

Documentation

Player, maker and converter of animations
for the Atari ST(e)

Table of contents

I Introduction

1) Overview

- MP_STE is a **player for various animations formats**:

- Quick Time MOV with sound
- Video for Windows AVI with sound
- GIF
- FLI/FLC/FLH from Autodesk animator
- FLM from Lexicor
- several other old Atari formats

- MP_STE can also **create animations** from a batch file

- Quick Time MOV with sound
- Video for Windows AVI with sound
- Extended FLM with sound

- MP_STE can **convert some formats to the Extended FLM** that is fully adapted to the ST(e) and allows you to get a **smooth replay**:

- QuickTime MOV with sound
- Video for Windows with sound
- FLI/FLC/FLH from Autodesk animator

2) Hardware requirements

- **MP_STE runs on**:

- Atari STe, Mega STe, Falcon, TT using the DMA Sound
- Atari STf, Mega STf using the Yamaha chip

- **The display** is either in

- ST Low 320×200 16 colors

- ST High 640×400 monochrome
- AlberTT graphic card on the Mega STE

➢ In addition to the integrated **sound system** (DMA or Yamaha), MP_STE can reproduce the sound via three sound cards:

- MV16
- ST Replay 8 or 16
- Psound parallel card

MP_STE can display **animations up to 640 pixels wide**, else results are unpredictable.

In ST LOW:

➔ If the width is greater than 320 pixels, the anim is reduced to its half.

➔ If the height is greater than 200 lines, only the center of the image is displayed.

In ST High:

➔ If the height is greater than 400 lines, only the center of the image is displayed.

On machines that **don't have the ST Low or ST High display** (*graphic cards, emulators, etc*) you can still use the FLM conversion feature or the creation from a BATCH file.

For example, Aranym can be used to perform fast conversions that will be copied to the STe.

3) Installation

You just have to copy those files into the s**ame folder**:

✔ MP_STE.PRG the program

✔ MP_STE.RSC the resource file

If you want to keep your **palette analysis** *(see **The MP_STE.PAL file**)* that can take some time to be computed, you can create an empty file:

✔ MP_STE.PAL

The player can also be installed as an **accessory**, then copy on your **boot disk**:

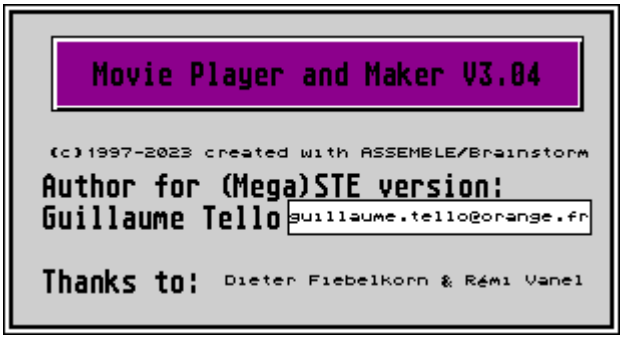
✔ MP_STE.ACC the program renamed as ACC

✔ MP_STE.RSC the resource file

The player is far better when a **hard disk** is available instead of floppies

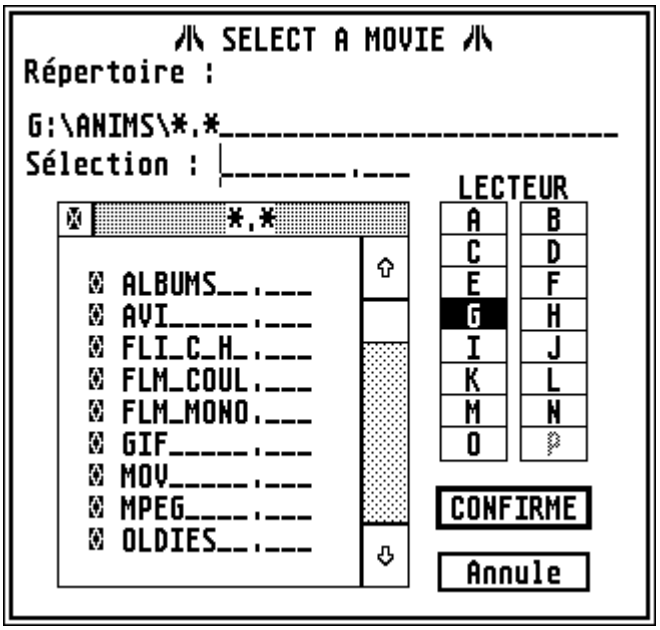
II Loading and playing a video

Run MP_STE and you'll get this information box:



Note : if you press SHIFT at start, then the mouse will never be hidden by MP_STE (useful to keep control with a debugger for example)

Just click and the file selector opens.



Here you can select a video.

MP_STE will analyze its header and give you the information about the video:

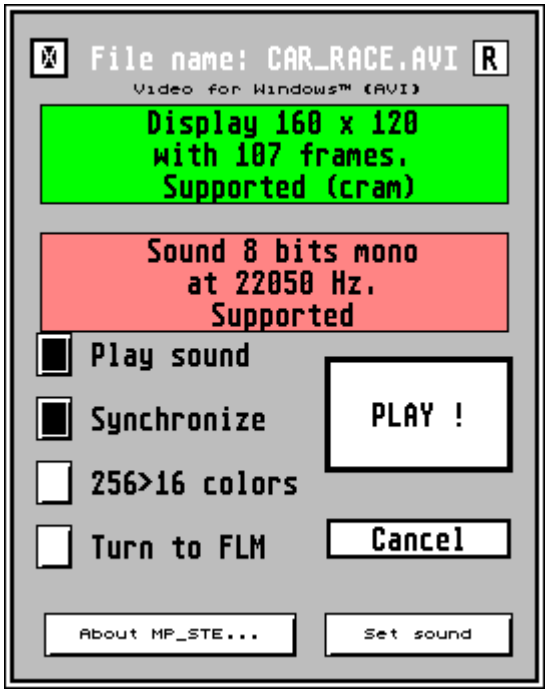
- ✔ video track info
- ✔ audio track info

If the analysis fails, then an alert box tells you that the format is unknown. *(see VII Supported file formats)*

In this example, the AVI format is detected (Video for Windows) and there are two tracks:

- ✔ video with 107 frames in 160×120
- ✔ audio with 8 bits mono sound at 22 kHz

Both audio and video are supported (this means that MP_STE knows the CODEC and can replay them).



Play button

If you clic on it, then you'll see:

for a short while and then the replay begins.

This is a choice I made for the player, the sound is fully loaded and converted to Atari format and then the replay begins.

Why? To let a maximum of CPU time to decode the images as there is no more work to do on sound.

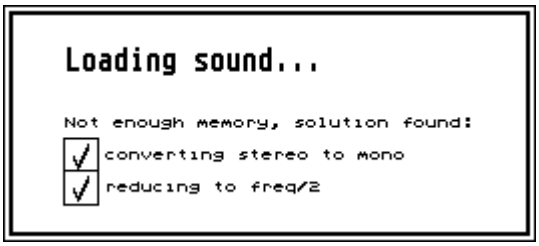
The advantage is that more images are displayed per second, the disadvantage is that the sound length is limited to the amount of RAM...

Sound reduction

In case the sound doesn't fit into memory, MP_STE tries to reduce it with two methods:

- ➔ converting stereo to mono
- ➔ reducing the frequency

So you'll see another box telling you what has been chosen



This is a view of the original AVI file as you could see it with M_PLAYER on a True Color display.

*(**M_PLAYER.PRG** is the version for 68030 that can use displays up to 32 bits)*

But, the ST(e) can't display this !

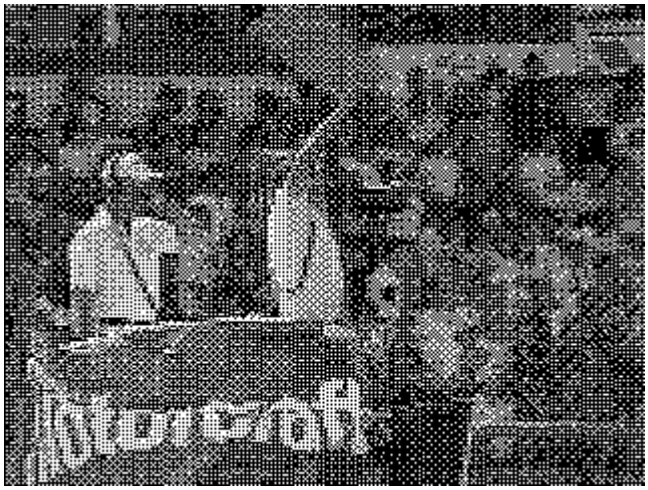
With the **ST Low display**, you'll get an image with 16 gray levels.

This will be the most common result for a majority of animations.

But, if the anim itself is limited to 16 colors (for example a AVI encoded in rle4 format, or most FLM files), you'll get the color display.

With the **ST High display**, MP_STE uses a Bayer filter to display the images.

This one is not the best for rendering, but the best for speed.



Control key

To stop prematurely an animation, use the Control key.

Play sound button

When this button is available, this means that the audio track is present. You can uncheck it to get a mute replay.

Synchronize button

When this button is available, this means that timing information for every frame can be used. The player does its best to follow the rate. If the **computer is too fast**, then a delay is respected before displaying the next frame.

If the **player is too slow**, then:

- for the AVI/MOV files, a system with key frames is present and allows a slow player to skip some frames to remain synchrony with the sound or with timing information
- for other formats with timing (FLI/C/H, FLM, GIF), the player doesn't wait and can lead to desynchronization.

If you uncheck it, all frames will be displayed at full speed.

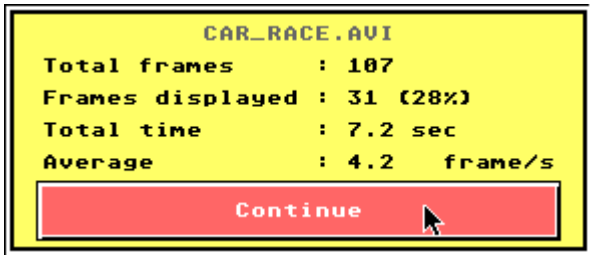
R button

When available, you can check it to repeat the animation.

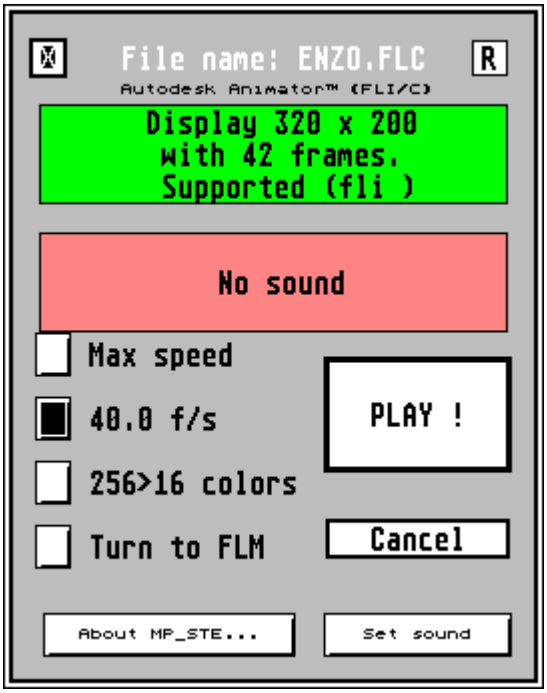
To stop the replay, press **Control**

Stats box

At the end of every animation, unless you used the Control key, you get this statistics box showing the performance of the player.



Autodesk FLI, FLC and FLH



For those files, no sound is included.

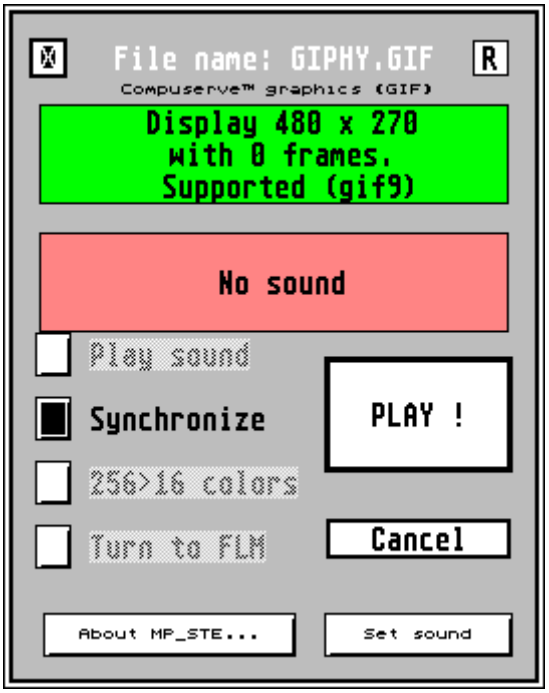
Buttons are labeled

➔ Max Speed

➔ xx.x f/s

This are radio buttons to allow you to respect (as far as possible) the original replay rate or to display with no timing information.

GIF CompuServe

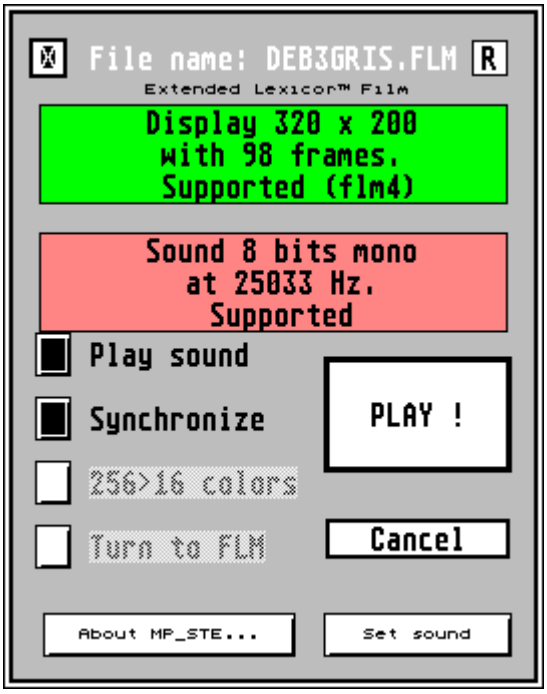


For those files, the header doesn't include the total number of frames, unless created by MP_STE that includes an extension for that.

So you'll mostly find a zero for the number of frames.

Again, there are not always timing information included. Take care, some GIF include a REPEAT flag. You can exit with Control, but if you want the statistics box, use SHIFT to stop at the end of one loop.

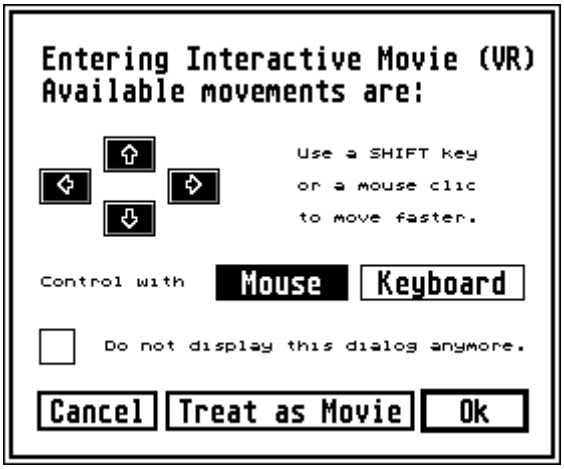
Extended Lexicor FLM



Lexicor has created a very good format to compress a video using directly the screen format of the ST. The decompression is made directly in the screen memory, this makes this format very fast.

I have extended this format to add information for timing and to support sound. To keep the spirit of Lexicor, the sound is directly at the DMA frequencies and can be loaded an replayed with no work at all.

Quick Time VR movies



These are special files, VR stands for Virtual Reality. You can move the point of view in your animation setting it with the keyboard or the mouse.

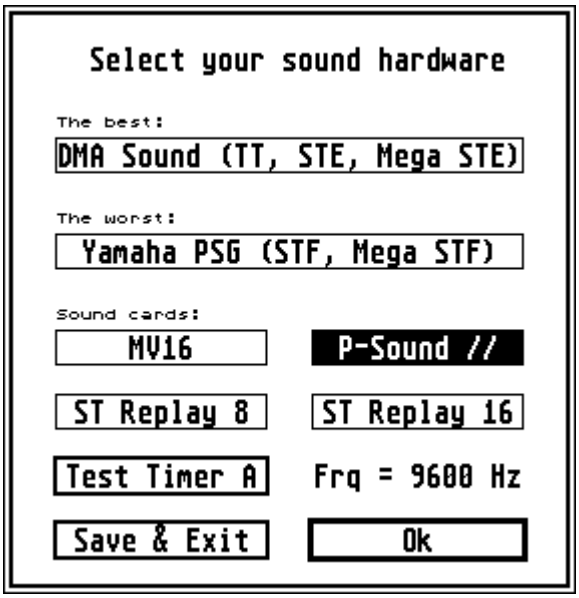
Only the VR/1 format is supported with fixed or animated cells. The VR/2 with panoramas is not supported in MP_STE, but it is in M_PLAYER.

You can select the control with **mouse** or **keyboard** (the arrow keys)

If you just want to see all the images without the interactive feature, clic on **Treat as movie**.

Else clic on **Ok**.

III Setting the sound



From the main dialog, you can Set the sound according to your system.

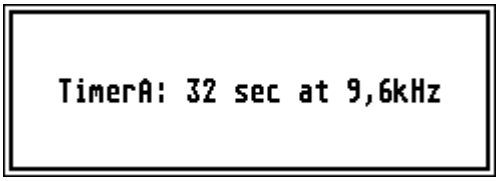
The best choice is the **DMA sound** if you have at STE or a TT because the CPU is not used at all.

The **sound cards** can be a good compromise on the STf, they require a part of the CPU time but the routine used by the Timer A is really short.

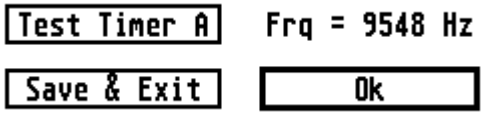
Last, the **Yamaha chip** can replay sounds, but the CPU usage is intense, the quality not so good.

If you want your choice to be permanent, click on **Save and Exit**. This will modify the RSC file.

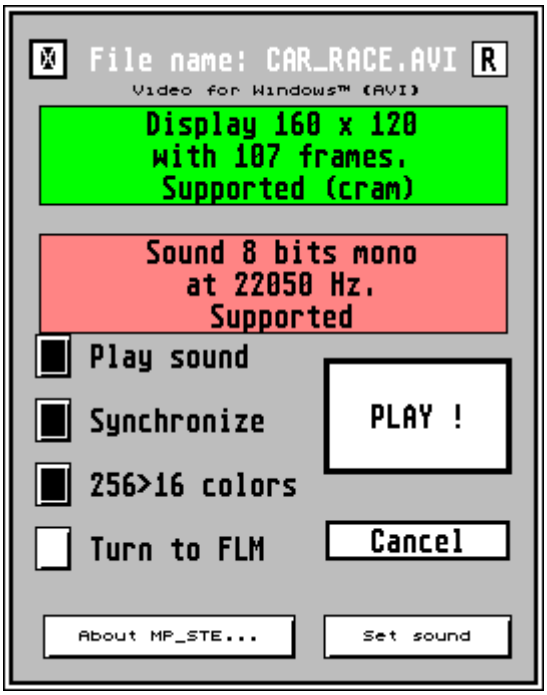
For any other system than the DMA, the TimerA is used to replay the sounds. On fast machines, the synchrony is correct but on slower ones, the computer doesn't really achieve the required 9600 Hz. To correct this, you can click on **Test Timer A**, this will run a 32 seconds test after which the actual frequency is displayed.



For example, here is what I get on my ST, again, if I want this value to be permanently used, I can click on **Save & Exit**.



IV Conversion from 256 to 16 colors



If the button **256->16** is available, then you can get a color display. But first, MP_STE needs to analyze the palette.

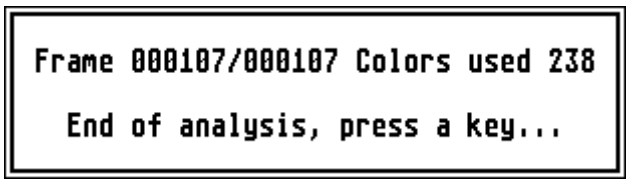
(this is available for DL, FLI/FLC/FLH and AVI/MOV with 256 colors)

So, this work is done in two passes:

First pass:

Check **256->16** and clic on **PLAY !**

A box appears telling you the current frame and how many colors out of 256 are actually used, at the end of the process, you get the result and have to press a key:



Second pass:

Back to the main dialog, check again **256->16** and clic on **PLAY !**

This time, the computer tells you that it's **compressing the palette** for a few seconds and the replay in color starts.

The result is not as good as the original, but shows a **correct approximation** of what it should look like.

Remember that **only 16 colors are used** for the whole animation!

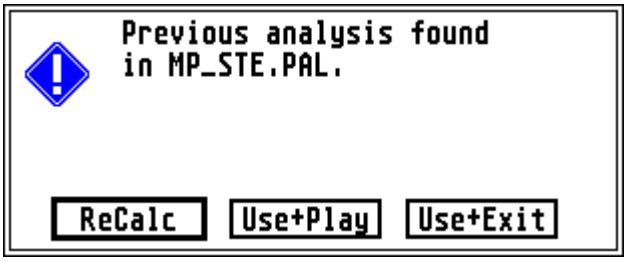
The algorithm for compressing the palette was given to me by Dieter Fiebelkorn (programmer of GemView).

The MP_STE.PAL file

On long animations, the palette analysis lasts a long time as MP_STE has to decode every frame. **If this file is not present** in the MP_STE folder, then the work is done every time it's needed.

But **if the file is present**, then MP_STE saves the result of every analysis in it and then, when the data is required again, it first looks in this file to rapidly retrieve the results.

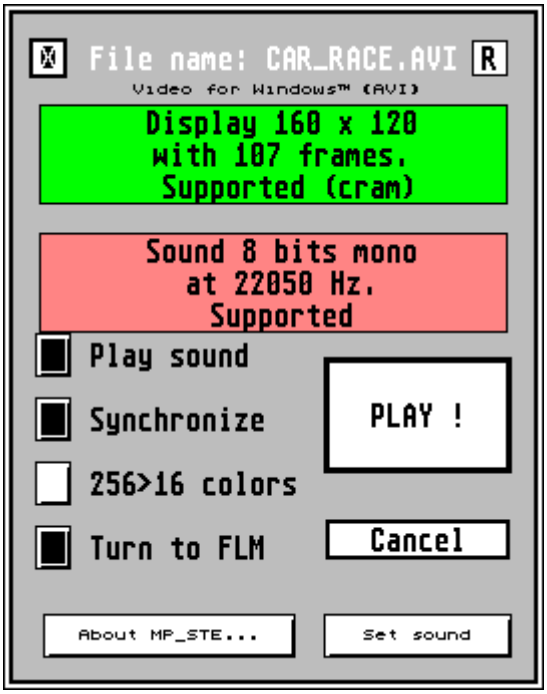
If this is the case, you'll see this alert box:



- Recalc: don't care and recompute everything. The PAL file is updated with the new data. Use this if your animation has been modified for example.
- Use+Play: get the data and use them to replay.
- Use+Exit: get the data and go back to the main dialog. This is useful if your goal was to "Turn to FLM" your animation.

MP_STE doesn't create the file, it's up to you to decide if you want to use this feature or not. To create an empty file, run a text editor and save with MP_STE.PAL name.

V Turning an animation to FLM

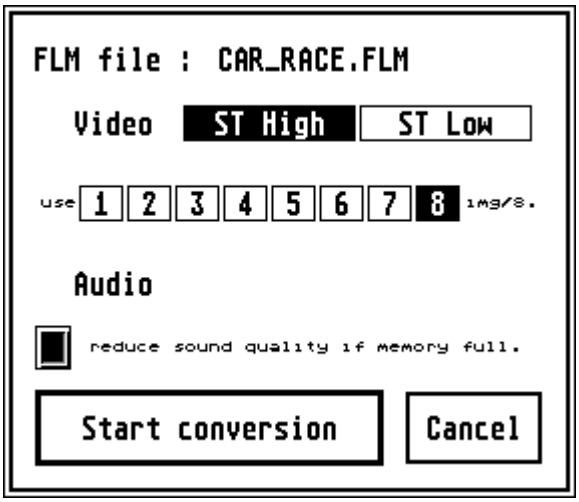


When **Turn to FLM** is available, you can convert your current video into a more suitable format for the Atari : the Extended Lexicor FLM.

Check **Turn to FLM** and clic on **PLAY !**

A **file selector opens** and asks you for the destination file name.

Ensure that the destination drive has enough free space! *MP_STE doesn't alert you if the disk is full.*



The **conversion box** opens. You can select the output format:

ST High: 640x400 monochrome. In this mode if the animation is less or equal to 320×200 then it is doubled to provide a better appearance with the Bayer filter.

ST Low: 320×200 16 colors or 16 gray levels. You can use this feature in conjunction with the 256->16 palette compression. This is discussed below.

Reducing the frame rate: in some cases:

- you want to reduce the final size of the animation
- the current animation, even converted, can't reach the frame rate

you'll want to reduce the number of images per second. Default is 8 images out of 8 (all of them).

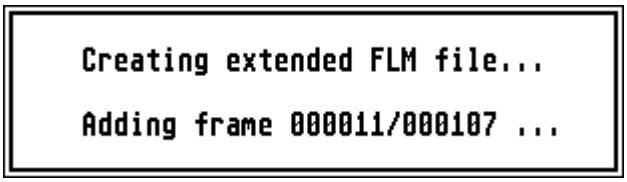
If you select 4 images out of 8, then you reduce the frame rate to the half.

Generally, each step down lowers the frame rate by 12,5%.

Again, as for replaying an animation, you can force the **sound quality** to be reduced to fit in memory *(see Sound reduction)*

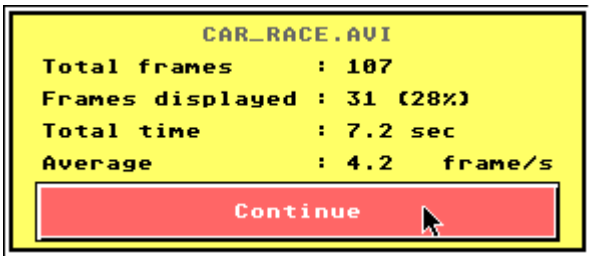
When everything appears to be correct, then clic on **Start conversion**.

A box appears with the current operation:

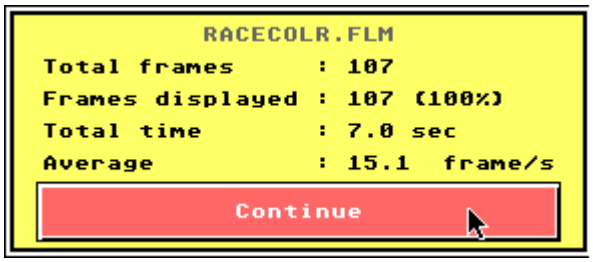


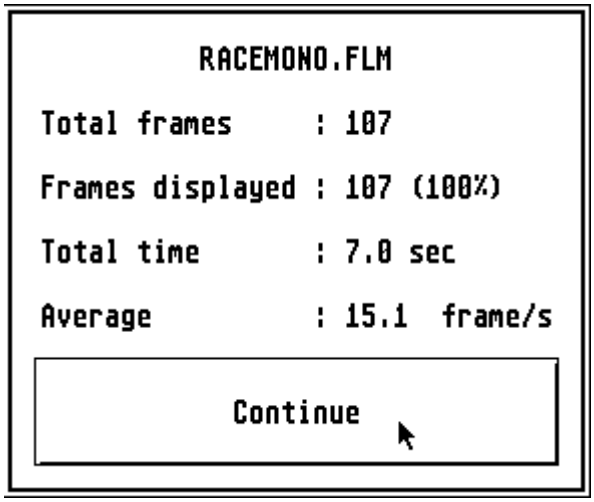
At the end, you get the **statistics box** and you'll be able to load and play the new created FLM file.

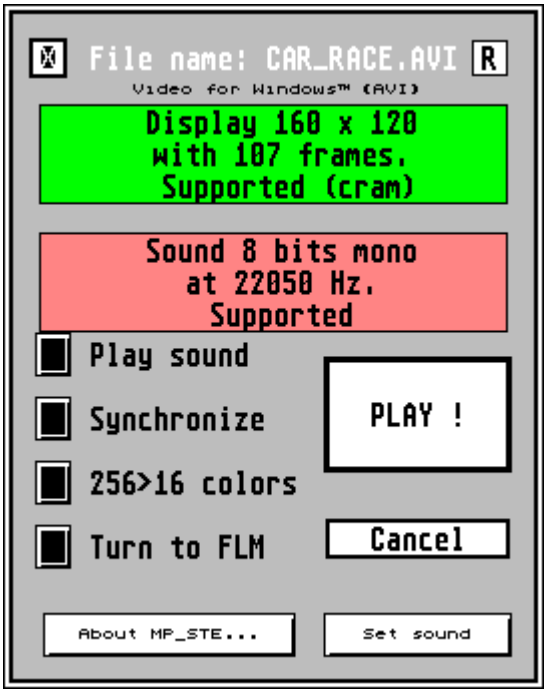
As an example, the AVI file presented above had a **poor replay quality** with only 28% of frames displayed:



When converted to FLM color or FLM mono, the **display is perfect and smooth** with 100% of frames displayed:







Conversion to FLM with palette compression

See at First pass: and run the first pass of the analysis. Then, back on the main dialog, check simultaneously **256->16** and **Turn to FLM** and clic on **PLAY !**.

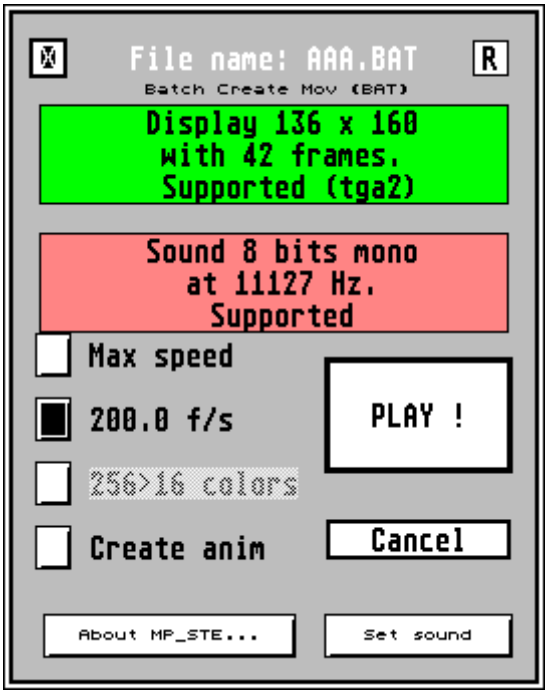
This time, the animation will be saved in colors!

Of course, the palette compression has no effect if you select the ST High conversion mode...

VI Creating animations

You can create animations using a BATCH file. This is a text file with the required information such as:

- ✔ the size and the number of frames
- ✔ the names of the individual images
- ✔ the name of an eventual sound or timing information
- ✔ the desired output type and output name



Load the batch file and you'll get this kind of dialog:

You can simply clic on **PLAY !** and the images are decoded and displayed.

If you are satisfied with that, you can check the button **Create anim** and then clic on **PLAY !**.

This time the output file is built.

The format is almost the same as in M_PLAYER. You can read, in **M_PLAYER.PDF**, the chapter:

V.3 Structure of a batch file, create it by hand.

But in MP_STE there are some limitations, as the CRAM compression is not included. So, here is what you can do:

- ✗ MOV RLE16 from TGA images
- ✗ MOV RLE8 or AVI RLE8 from X-IMG files
- ✗ GIF animated from GIF images
- ✗ FLM from Degas or Neochrome images

This chapter, in M_PLAYER.PDF, may be useful too:

IX EASY_BAT.PRG (discontinued)

VII Supported file formats

Here is the list of what MP_STE can read.

QuickTime *.MOV (VR1 Objects supported)

Video codecs : CVID, RLE1, RLE2 (gray and color), RLE4 (gray and color),
RLE8 (gray and color), RLE16, RLE24, RLE32, SMC8 (gray and color)
RAW1, RAW2 (gray and color), RAW4 (gray and color),
RAW8 (gray and color), RAW16, RAW24, RAW32, RPZA (15 bits)
WRLE (256 colors), MSVC8 (gray and color), MSVC16
YUV2, YUV9, YVU9
Audio codecs : TWOS, RAW (8/16 bits, mono/stereo)

Video for Windows *.AVI

Video codecs : CVID, CRAM16, CRAM8, MSVC16, MSVC8
RLE8, RGB8, YUV9, YVU9, IV32
Audio codecs : TWOS, RAW (8/16 bits, mono/stereo)

Extended Lexicor Film *.FLM

Video codecs : ST Low, ST High
Audio codecs: DMA 8 bits mono/stereo, 12,5kHz, 25kHz, 50kHz.

CompuServe *.GIF

Video codecs : GIF87, GIF89

Video Master *.FLM

Video codecs : VMAS 160×100 16 colors
Audio codecs : 8 bits mono, any frequency

CD Stratos Magazine *.CDH, *.CDL

Video codecs : ST Low, ST High

Audio codecs : DMA sound 8 bits mono 12,5kHz

Old ST Low Atari animations

- *.SEQ (Cyber Paint)
- *.PI1 + *.DLT (Cyber)
- *.FLM (Kinetic Microsystems)
- *.FLM (Lexicor)

Old PC animations

- *.FLI/FLC/FLH (Autodesk Animator)
- *.DL (types 1 and 2)

Batch Files *.BAT

For slide shows or creation of animations.

Images : TGA2, TGA10, 16 or 24 bits, uncompressed or RLE

GemXIMG 256c, GIF, PI1, NEO

Sound : AVR or WAV, uncompressed.