

Fixing synchronisation problems using a video editor

One of the most common problems with desktop video is when video and audio gets out of synch. In the old days we used to extract the audio content, decompress it into a wave file, edit the wave file, recompress it and combine it with the video again. This solution has two distinct drawbacks: it's very difficult to determine how to exactly adjust the decompressed audio file to obtain precise synchronisation, and the sound quality is seriously degraded by the recompression of the audio.

This text describes a much simpler and faster way to solve most of these problems without these drawbacks, using cheap or even free software.

Pro-active solution

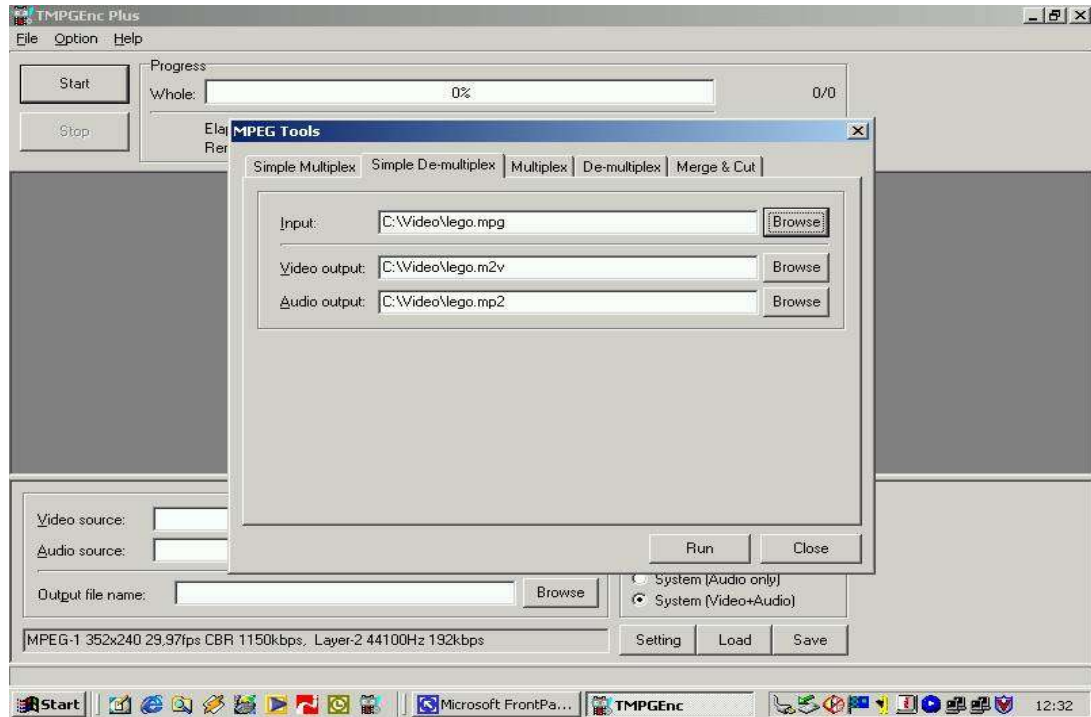
First a general suggestion: I believe that most synchronisation problems occur because of incompatibilities between video and audio codecs, i.e. the small pieces of software that helps your Windows system encode and decode digital audio and video. Whenever you install a sound or video editor, a player program or even a CD writing program, you install more and more of these little useful but complicated things. The more codecs you have installed in your system, the more conflicts you can expect between them. There may be version conflicts between different versions of the same codec, or your applications may be forced to use codecs different from the ones they came with - and have been tested with.

To avoid synchronisation problems, you may consider running your most important video applications in separate environments. In my case, I used [PartitionMagic](#) to create three separate hard disk partitions (Windows installations) for capture, edit/burn and sound editing. Since each of these codec-sensitive program now runs in its own isolated environment, using only the codecs it came with, results are much more predictable and support issues are much easier to handle. The overhead of rebooting into different Windows partitions is negligible compared to the problems you avoid.

Corrective solution

To fix synchronisation problems in a simple way, you need a video editor with at least one additional, separate sound track (e.g. for voice or music overdubs). I use [Ulead VideoStudio](#), which I consider one of the best programs around. It's also fairly cheap.

1. Start by de-multiplexing the video and audio of the original MPEG file into separate temporary files. Use [TMPGEnc](#) or any other suitable program. The de-multiplexing operation should create one m1v/m2v file for the video, and one mp2 file for the audio.



2. Open the m1v/m2v video file in your video editor. Ulead VideoStudio doesn't list m1v/m2v files, but you can rename the file and give it an mpg extension.



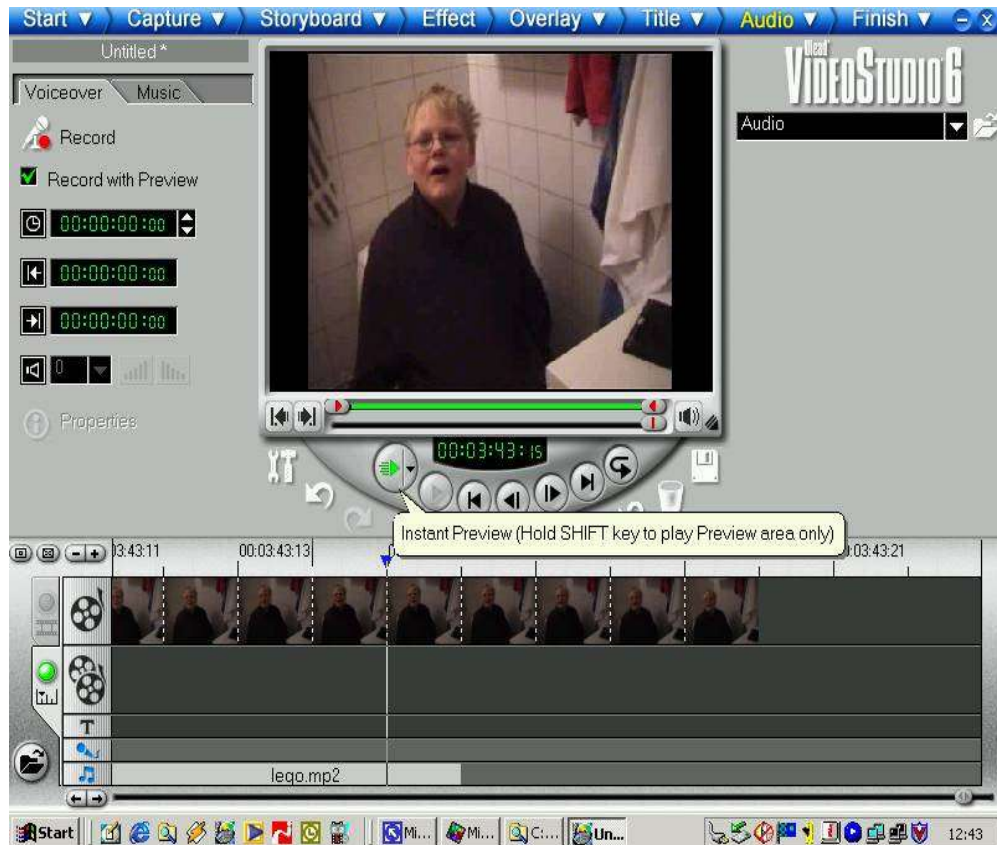
3. Insert the mp2 audio file into one of the separate sound tracks of the storyboard (I normally use the music track in Ulead VideoStudio).



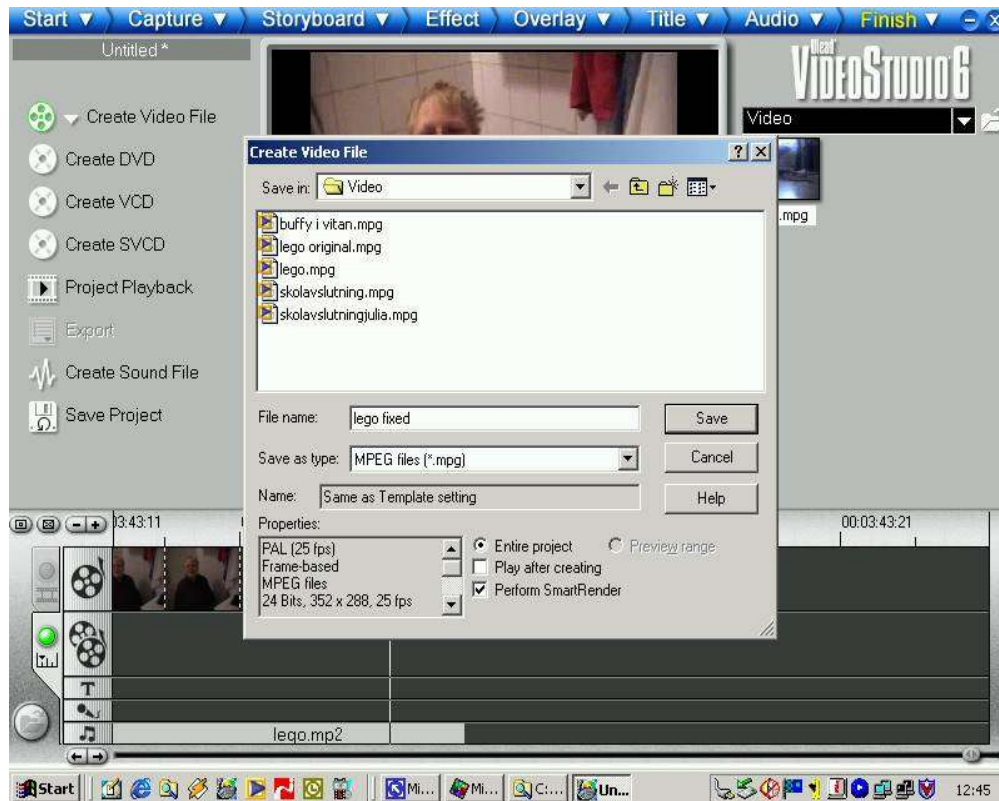
4. If the audio is ahead, slide the inserted audio file in the separate sound track to the right in the timeline, so that it starts after the video. If the audio is behind, trim the beginning of the audio file and align the remaining audio segment with the first video frame. In my setup a good starting point is to adjust the starting point for the audio file to 0:0:0:4 (four frames) and slide the entire audio file to the start of the timeline. The exact offset may be difficult to locate at first, but after some practise you will know where to start your tests.



5. Place the editor in Preview mode and start playback. The editor will play the video from the (soundless) video track, adding the sound from the separate sound track. Verify that audio and video is now in synch. If it isn't, stop the preview and adjust the sound track again as described in step 4, then try again. Repeat this process until the audio is in synch with the video.



6. Video editors normally provide a "Generate" function that allows you to render a new video file directly from the storyboard. With the adjusted audio track, the video and audio will be in synch in the video you render, but most editors (including Ulead VideoStudio) will decompress the different audio tracks before they are mixed together. The composite audio track is then recompressed and added to the video. This recompression degrades the sound quality of the audio. If this is not a problem for you, you may now simply render the final video, and your synchronisation problem is solved.

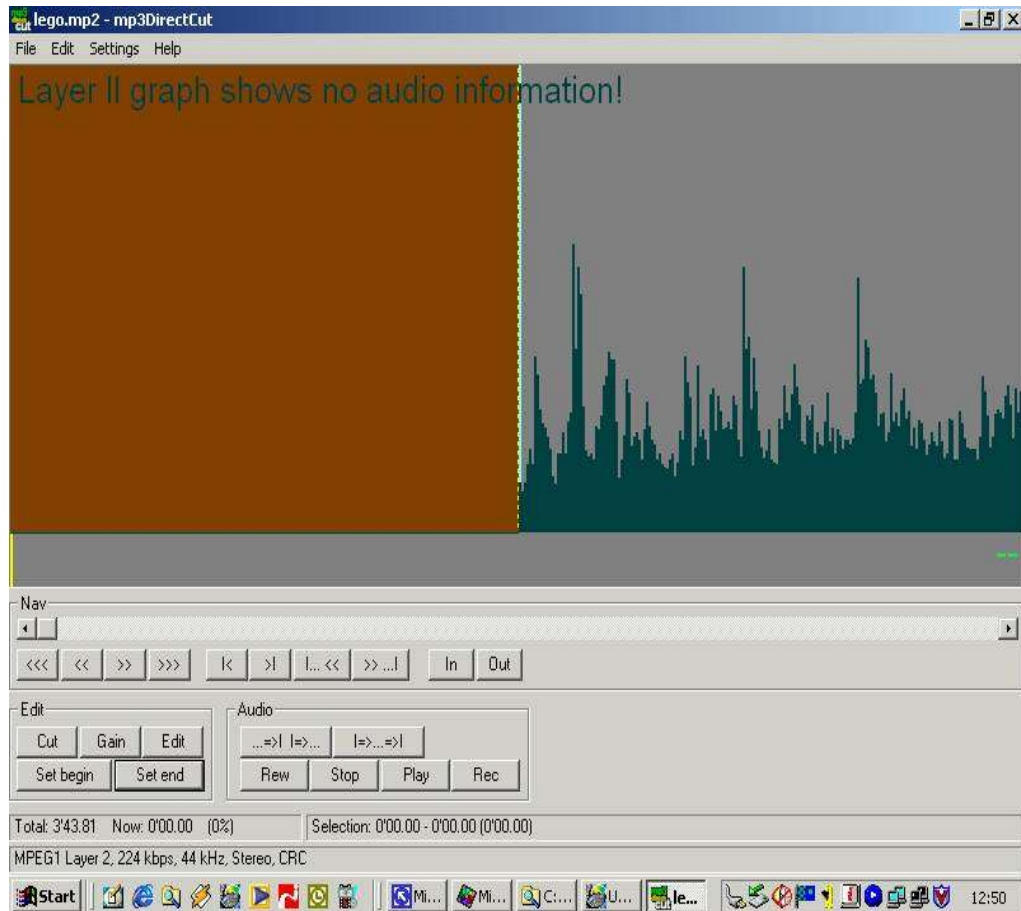


Avoiding sound quality degradation

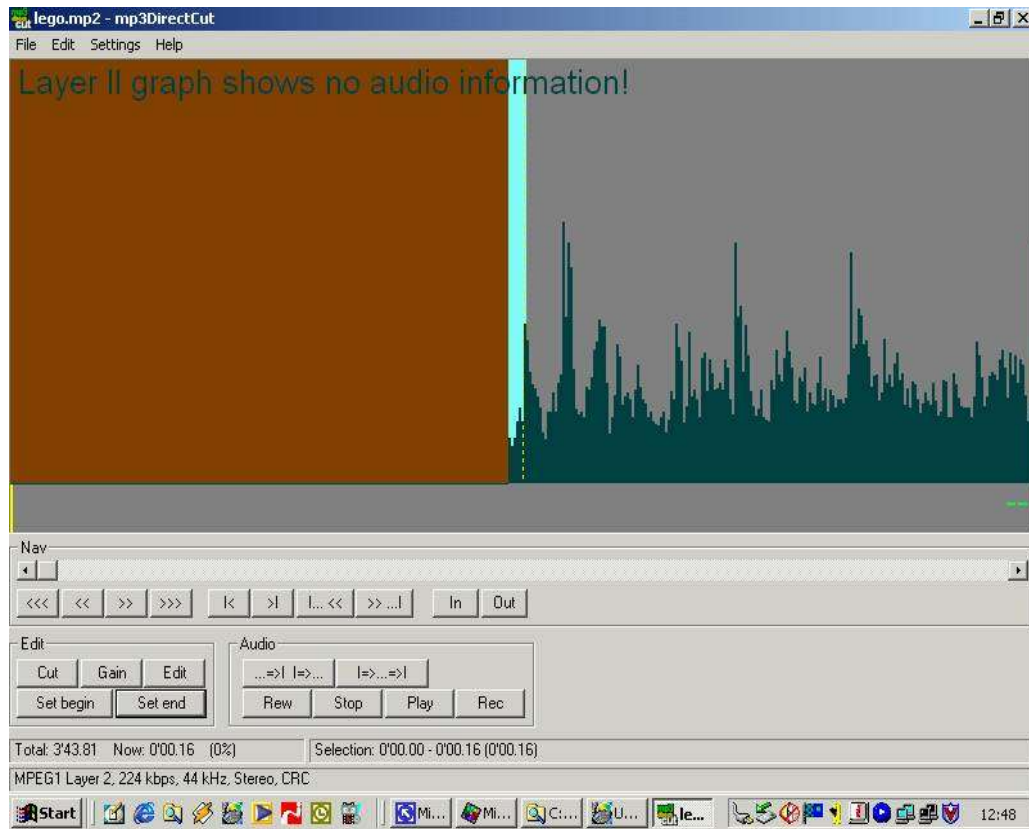
To maintain the original sound quality, you may instead trim the mp2 file directly, then combine it with the video without recompression. The MP3DirectCut editor available at <http://www.rz.uni-frankfurt.de/~pesch/> provides a fast and simple way to manipulate mp2 and mp3 files without recompression.

One problem you will encounter is that video editors normally use a h:m:s:f notation, where "f" is the frame number. MP3DirectCut uses the h:m:s:d notation, where "d" is the decimal portion (e.g. "0:0:0:5" for half a second). Before you edit the sound file, you need to convert the exact offset between the two notations. Use the "h:m:s" part exactly as it appears in the video editor. Divide the number of frames displayed in the video editor (e.g. 4) with the frame rate of your video (e.g. 25 for PAL or 30 for NTSC) to obtain the decimal equivalent. In my example above, 0:0:0:4 in Ulead VideoStudio becomes 0:0:0:16 (4/25) in the MP3DirectCut editor.

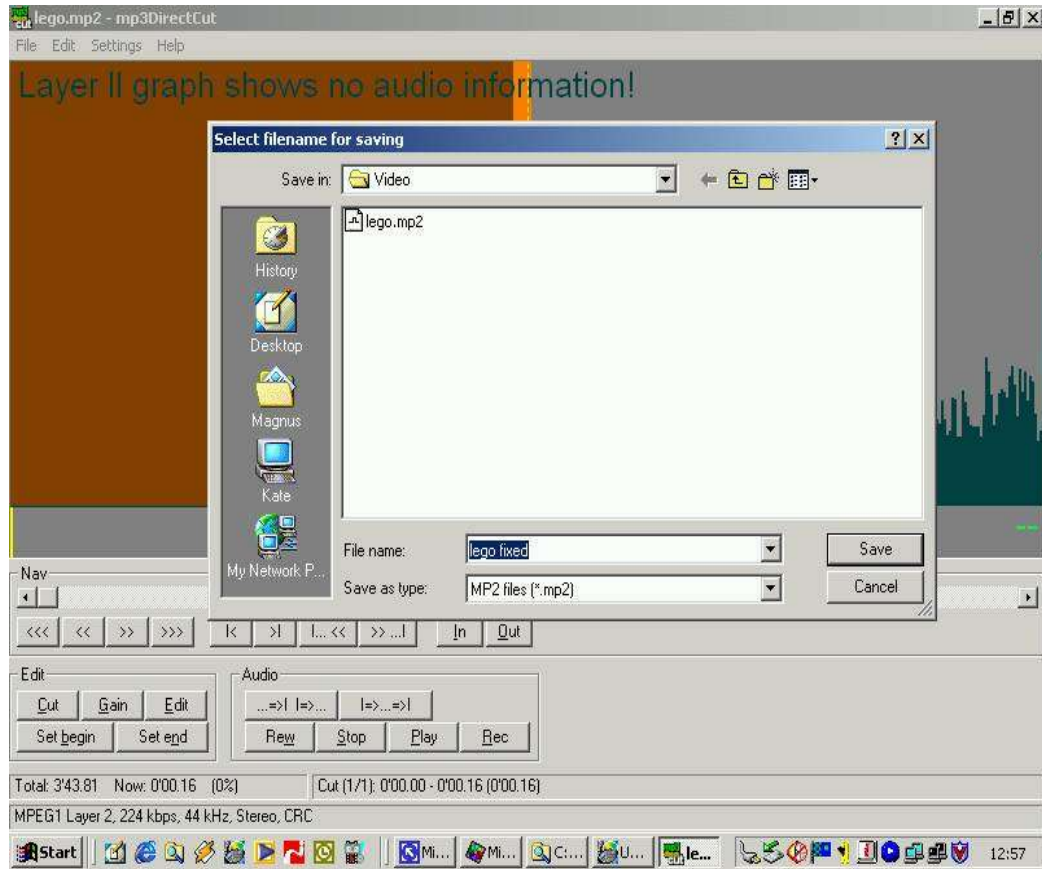
1. Open the mp2 audio file in MP3DirectCut.



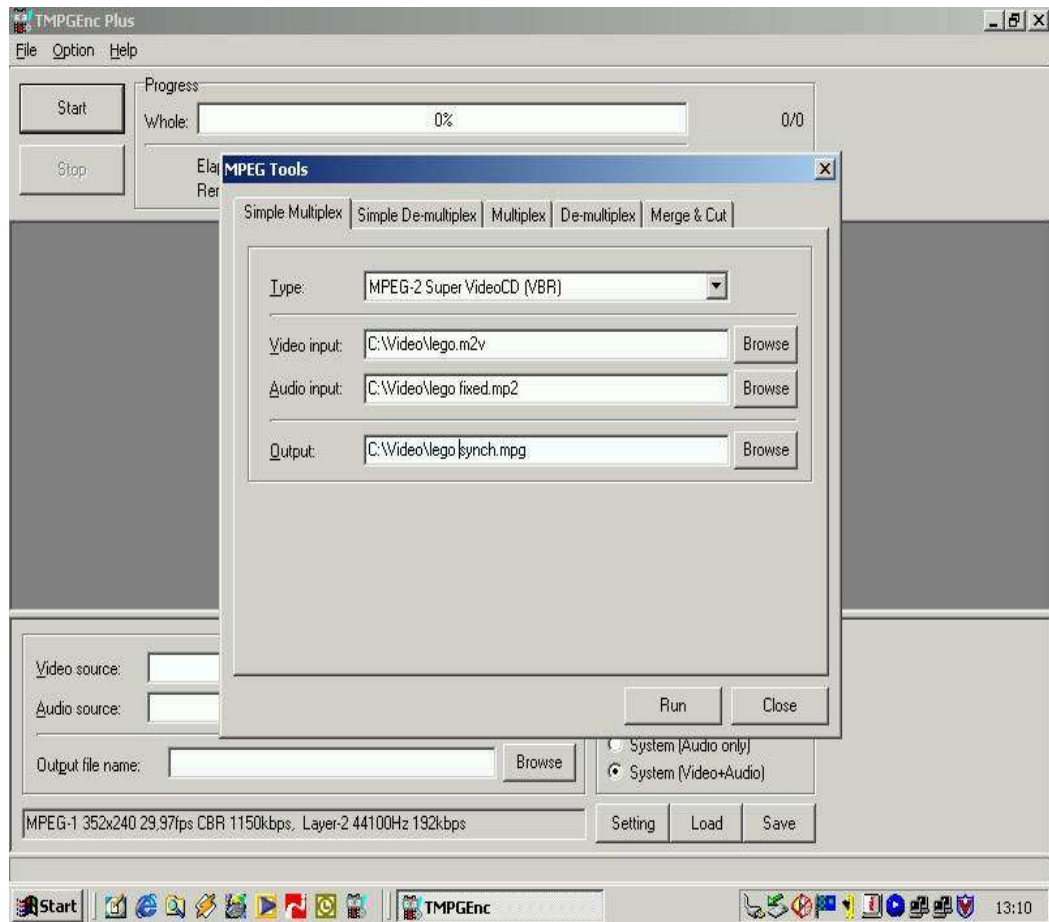
2. If the audio was behind, set the end point to the position calculated above, then cut the beginning of the video file. If the audio was ahead, copy a silent part with the correct length and insert it at the beginning of the audio file to delay the rest of the audio.



3. Save the edited mp2 file to disk.



4. Mux the edited mp2 file with the video-only file (remember to set the correct stream type) to obtain a final video with exact synchronisation, without recompression or sound degradation.



A final tip

In the worst case, you will achieve synchronisation for one part of the video but the sound will then gradually start to drift out of synch again. In this case you may consider splitting the original video file into two or more separate files before you correct the synchronisation problem separately for each file using the instructions above. Since this can be very time-consuming, you may instead decide to implement the pro-active solution described at the beginning of this tutorial. With an isolated environment for each multimedia application, you may never have a synchronisation problem again.

This document was last updated on 07/01/02