

Working way to make video from images in C#

Asked 15 years, 10 months ago Modified 11 years, 1 month ago Viewed 32k times



19



Does anybody have a known reliable way to create a video from a series of image files? Before you mod me down for not searching for the answer before posting the question, and before you fire off a simple message like "use FFMPEG," read the rest of this message.

I'm trying to create a video, it doesn't matter too much what format as long as it's widely supported, from a series of images (.jpg, .bmp, etc.). My platform is Windows Server 2008, 64-bit. If I can make the video from within my C# program, that's great, but I'm not averse to writing a series of image files to a directory and then firing off an external program to make a video from those images.

The only constraints are: it must work on my Windows Server 2008 system, and be scriptable. That is, no GUI programs that require operator intervention.

I found a number of similar questions on StackOverflow, and have tried several of the solutions, all with varying degrees of frustration and none with anything like success.

FFMPEG looks like a great program. Maybe it is, on Linux. The two Windows builds I downloaded are broken. Given this command line:

```
ffmpeg -r 1 -f image2 -i jpeg\*.jpg video.avi
```

One of the builds reads the images and then crashes due to data execution prevention. The other reads the first file and then spits out an error message that says "cannot find suitable codec for file jpeg/image2.jpg". Helpful, that. In any case, FFMPEG looks like a non-starter under Windows.

One answer to a previous posting recommended [Splicer](#) . It looks like pretty good code. I compiled the samples and tried to run, but got some cryptic error message about a file not found. It looks like a COM class isn't registered. I suppose I need to install something (DirectShow, maybe, although I thought that was already installed?). Depending on what's required, I might have a difficult time justifying its installation on a server. ("What? Why do you need that on a server?")

Another answer suggested the [AviFile](#) library from Code Project. That looks simple enough: a wrapper around the Windows AviFile subsystem. Except that the AVI files the package creates appear to have all of the frames, but only the first frame shows when I play the AVI in Windows Media Player. Well, that and if you try to create a compressed video, the program throws an exception.

So, I'm left wondering if there is a good, reliable way to do what I want: on a Windows system, create an AVI or other common video file format from a series of images, either through a .NET API or using an external program. Any help?

c# video ffmpeg

Share Improve this question Follow

asked Feb 11, 2009 at 23:00



Jim Mischel

134k ● 24 ● 194 ● 362

Try your ffmpeg command without -f image2, or with -f jpeg, i.e. 'ffmpeg -r 1 -i jpeg/*.jpg video.avi' or 'ffmpeg -r 1 -f jpeg -i jpeg/*.jpg video.avi' – [configurator](#) Feb 11, 2009 at 23:22

@configurator: without -f image2, I get the same results. With -f jpeg, I get "Unknown input or output format: jpeg". – [Jim Mischel](#) Feb 12, 2009 at 0:07

5 Answers

Sorted by: Highest score (default)



After working with it a while and taking a look at x264 and VideoLan, I went back to Splicer. It turns out that the cryptic error message was due to an error in my code.

11



It looks like Splicer will do what I want: programmatically create videos from a series of images.



Thanks to all who responded.



Share Improve this answer Follow



answered Feb 12, 2009 at 21:21



Jim Mischel

134k ● 24 ● 194 ● 362



You can use VideoLan and I'd recommend it.

6



I've had direct experience in a C# application with VideoLan doing these two things:

1. Embedding a VLC viewer in my C# application (there are 3-4 C# "wrappers" for the VLC viewer).



2. Using vlc.exe in a separate Process by sending it command-line arguments.



The fact that VideoLan has a command-line interface is a great thing. And VLC supports a command-line option that disables any visual element; so the VLC GUI doesn't pop up and neither does a command-line window. Thus, in a C# application

you can farm out the video-related work to the VLC client. C# has the Process class which can manage your vlc.exe instances for you. It ends up being a pretty neat solution.

Share Improve this answer Follow

answered Feb 12, 2009 at 4:51



Chris Holmes

11.6k ● 12 ● 52 ● 64



1



The handling of video on Windows (encoding and playback) is a real quagmire. And .NET isn't going to help you at all - you're always going to be either PInvoking/COM-interoping native APIs or using a .NET wrapper that someone else has built to do the same.

Anyway, I can't say I've used it, but a good bet seems to be the [x264 library that is part of the videolan project](#). That gives you a self-contained C library that you can PInvoke and you don't have to worry about installing and DirectShow garbage on your server. It'll generate H.264 format video which is playable in Flash as I understand it.

Share Improve this answer Follow

answered Feb 12, 2009 at 0:54



U62

4,363 ● 1 ● 26 ● 39



0



I've used [mplayer](#)'s mencoder to create AVI (with codecs msmpeg4/mp3, readable by default with Windows Media Player) from TGA files on linux like this:

```
mencoder mf://*.tga -mf fps=25:type=tga \  
-audiofile /tmp/test.mp3 -oac copy \  
-of lavf -ovc lavc \  
-lavcopts vcodec=msmpeg4v2:mbd=2:mv0:trell:cbp:last_pred=3:vbitrate=3000 \  
-o /tmp/test.avi
```

I know mplayer does have a Windows version.

Share Improve this answer Follow

answered Feb 12, 2009 at 8:55



Tometzky

23.9k ● 5 ● 63 ● 78



-2

Instead of just going through all the hassle, you could copy all the images that you want to be included in your video to the clipboard and have it imported into Windows Live Movie Maker (dynamically, of course!) and have it compile for you!



All of this CAN be done through code, but this way, you don't have to create it yourself. As you said that the platform was Windows, and ALL Windows comps have Windows Live Movie Maker installed, so it shouldn't be a problem for anyone using your application unless

a) They uninstalled Windows Live Movie Maker

b) They are using a computer running on a Windows Version more than 8 or 9 years old...

Share Improve this answer Follow

answered Nov 29, 2010 at 23:07



[1-800-RUGS](#)

1

-
- 1 @Rugs: Not all Windows computers have Windows Live Movie Maker installed. It certainly isn't installed by default on Windows Server 2008, and lots of developers use that for their development machines. Secondly, your solution could work if you had some automated way of adding the thousands of images to the clipboard and importing them one at a time to Movie Maker. But mostly I'd reject this solution because I have a program creating the images dynamically and importing them one-by-one into Movie Maker would take an incredibly long time. – [Jim Mischel](#) Nov 29, 2010 at 23:22
-