

Re: How to play audio and video from two different sources

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.directx.video/2006-03/msg00>

- *From:* "DevX" <i.hate.all@xxxxxxx>
 - *Date:* Wed, 22 Mar 2006 15:49:03 +0100
-

Hi Alessandro (sorry I have switched "nome e cognome"... I'm the italian guy you helped a month ago in my post "Help regarding writing a Source Filter").

Thanks for the info, I saw the statement with `IMediaSample::SetTime()`. Now I have the info for proceeding writing the second pin for the audio.

Ciao

Dev

"Alessandro Angeli [MVP::DS/MF]" <nobody@xxxxxxxxxxxxxxxxxxxxxx> wrote in message news:#4m9jubTGHA.4600@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

DevX wrote:

Hi Angelo,

I don't know who Angelo is, but hi to you to :-)

How I set the timestamp on the samples? My samples are only a bunch of bytes representing a YUV buffer or a WAV sample, there is some form of DirectShow pre-processing I must do? Currently the only timing facilities I have on my filter are the same found on the BouncingBall, in the `Notify()` function were I set the repeat time of the frame.

If you copied `CBallStream::FillBuffer()`, then you are calling `IMediaSample::SetTime()` to set the sample's timestamp. The stop time is not really needed, but the start time is important: as long as you set the correct start time to each audio and video sample, the renderers will keep the

Re: How to play audio and video from two different sources

streams in sync.

How to calculate the start times depends on how you produce the data, but usually the start time starts at 0 (CBallStream::OnThreadCreate()) and for each sample S, $\text{start_time}[S] = \text{start_time}[S-1] + \text{duration}[S]$.

With video, $\text{duration}[S]$ is usually a known value (VIDEOINFOHEADER::AvgTimePerFrame), if the frame rate is constant.

With PCM audio, $\text{duration_in_secs} = 8 * \text{buffer_size} / \text{wBitsPerSample} / \text{nChannels} / \text{nSamplesPerSec}$ or $\text{duration_in_secs} = \text{buffer_size} / \text{nAvgBytesPerSec}$ (since, for PCM audio, $\text{nAvgBytesPerSec} = \text{wBitsPerSample} * \text{nChannels} * \text{nSamplesPerSec} / 8$).

Thanks a lot for the help and the patience.

You're welcome.

--
// Alessandro Angeli
// MVP :: DirectShow / MediaFoundation
// a dot angeli at psynet dot net