NAME

Mpegsmux - MPEG 1 system stream multiplexer

SYNOPSIS

```
mpegsmux <global options> <per-stream options>
```

In almost all cases you will only need to use a small number of options to make this command useful. For example, to create a generic multiplexed stream named `foo` of an audio and video sequence named `seq.ampg` and `seq.vmpg`, the command below will do that:

```
mpegsmux stream=foo audio=seq.ampg video=seq.vmpg
```

If instead, you need to produce this for CD-i or Video CD 2.0, then simply add the format specification, like these:

```
mpegsmux stream=foo format=cdi audio=seq.ampg video=seq.vmpg
mpegsmux stream=foo format=videocd audio=seq.ampg video=seq.vmpg
```

The most interesting options are those that allow you to concatenate sequences (`videos=` and `audios=`) or add still pictures to Video CD 2.0 and CD-i output (`stills=`). All the other options should be considered as knobs and dials to "tweak" your sequence into something slightly different.

GLOBAL OPTIONS

stream= *file* — Name of the system stream output file

verbose — If this keyword appears, `mpegsmux` will provide verbose output about the contents of each sector. It also includes output about how the decoder would decode the stream. This can be very useful if you have problems with "buffer underflow" or "buffer overflow". Note that this only works for CD output, not generic multiplexing.

intialscr= *time* — Set the initial SCR value to "time" seconds, default is 0.

muxrate= *bps* — Set the muxrate to "bps" bit per second. If not specified 1.01 times the sum of the elemetary stream rates inflated by the packetization overhead plus the minimum pack header and system header data rates will be used.

ratebound= *bps* — Set the bound on the mux rate given in the system header; defaults to muxrate.

pad=0/1 — A "1" implies a padding stream to achieve a data rate exactly equal to the mux rate. In this mode the number of bits in the output file is proportional to the presentation time. This mode should be used if the target medium is a synchronous communication channel or storage device. If "pad=0" (default), then pack header containing a time stamp are used to terminate "gap in time" found between packet of the bitstream. This mode is useful for asynchronous networks and recording devices.

packtime= *time* — Will force a pack header to be omitted at least every "time" seconds. A pack header will also be sent following any channel idle period (regardless of padding). Defaults to 0.3 seconds for normal operation. This option is ignored for compact disc operation.

headertime= *time*
: Will for a system header to be emitted at least every "time" seconds. This rate also controls the frequency of sending buffer sizes in each elementary stream. Defaults to 0.5 seconds for normal operation. This option is ignored for compact disc operation.

maxbehind= *sec*
: Maximum number of seconds between input and output (defaults to 2 maximum sized packets). This option is ignored for compact disc operation.

audiolock=0/1
: Sets the audio lock flag in system header (default=1)

videolock=0/1
: Sets the video lock flag in system header (default=1)

format= *type*
: Sets the stream encoder mode. The type may be any of `mpeg`, `cdi`, videocd, `vcdseg`. Without this flag the default is standard mpeg (ISO/IEC 11172-1). The type `cdi` means CI-Interactive or "Green Book"; `videocd` means Video CD 2.0 or "White Book"); `vcdseg` mean Video CD 2.0 segment-play. For any of the types `cdr`, `videocd` or `vcdseg`, the options described under GLOBAL COMPACT DISC OPTIONS may be used.

GLOBAL COMPACT DISC OPTIONS

filenbr= *file*
: Specifies CD subheader file number. If not specified, then only the system multiplexed stream with 2324 byte video "packs" and 2304 byte audio "packs". If specified, the output will be the "packs" described above inside of 2336 byte CD sectors. Essentially the sectors begin with a subheader, followed by data, followed by EDC code. More detail can be found the the appropriate standards.

channbr= *chan*
: Specifies CD subheader channel number. If not specified, a default is chosen appropriate to the standard. This is only meaningful if `filenbr=` is specified.

pause= *nsect*
: Specifies the number of sectors in pause region of CD track [def=150]. This is only meaningful if `filenbr=` is specified.

fmsect= *nsect*
: Specifies the number of sectors in front margin of CD track [def=15]. This is only meaningful if `filenbr=` is specified.

rmsect= *nsect*
: Specifies the number of sectors in rear margin of CD track [def=15]. This is only meaningful if `filenbr=` is specified.

stills= *file*
: Specifies a file containing a list of still frames and timecodes. The file format is described under STILL PICTURE FILE FORMAT

tcfps= *fps*
: This specifies the frames/second for still picture timecode parsing. If unspecified, it defaults to 30fps.

endtime= *time*
: Specifies the last packet time for the finished, multiplexed stream. When producing sectors for a compact disk, the sectors must continue to be delivered until the last presentation time stamp has been delivered. This parameter allows you to specify an end time beyond the last presentation time stamp.

For example, if the last still picture in a Video CD segment play occured at time 00:01:30:00 (or 90 seconds), and you wanted it to be displayed until 100 seconds, then the parameter `endtime=100` would do that. This argument is ignored if any elementary stream has a presentation time stamp that is later in time.

PER-STREAM OPTIONS

audio= *file*
: Introduces an audio stream from file "file". Up to 32 distinct audio channels can be multiplexed and they are denoted by stream IDs of 0xC0, 0xC1, ... in the order they appear.

audios= *file*
: Same as audio= *file*, except that *file* is a list of audio sequences with optional time codes and EOR markers. See SEQUENCE LISTS below.

video= *file*
: Introduces a video stream from file "file". Up to 16 distinct video streams can be multiplexed and the are denoted by stream IDs 0xE0, 0XE1, ... in order.

videos= *file*
: Same as video= *file*, except that *file* is a list of video sequences with optional time codes and EOR markers. See SEQUENCE LISTS below.

data= *file*
: Introduces a data stream from file "file". Up to 16 different data streams can be multiplexed and they are enumerated by IDs 0xF0, 0xF1 ...

packetsize= *bytes*
: Gives the packetsize for this stream. This number includes the 6 byte packet overhead plus the optional fields of PTS, DTS and STD buffer size. If one of the optional fields is included, the number of payload bytes is decreased to keep the packet size constant.

bufsize= *bytes*
: Sets the STD buffer size for this stream. The default is 4096 bytes for audio & data and the VBV size plus one packet size (for multiplex jitter) for video.

bitrate= *bps*
: Sets the elementary stream rate (bits per second). This is a required parameter for a data stream; illegal for video and audio streams (smpegcomp extracts the bitrate from the audio and video stream syntax).

ausize= *bytes*
: Set the access unit size (and time) for a data stream; illegal for video and audio streams (this paramter is determined by partially decoding elementary audio and video streams).

aualign=0/1
: This flag specifies that a new access unit will be placed in the beginning of a new packet.

initialocc= *fract*
: This set the initial buffer fullness for an elementary stream.

maxbufsize= *bytes*
: Specify the maximum buffer size for this stream (defaults to the actual buffer size).

starttime= *sec*
: Specifies initial presentation delay in seconds. The default is 0.0. The stream synchronization procedure is as follows. First, additional delays for each stream are calculated without regard to 'starttime' so that the initial presentation times are concurrent at the largest presentation time stamp (PTS). Next, each stream is further delayed from that point by the ammount of its

'starttime'. Note that this option is ignored if the elementary streams are specified with `videos=` or `audios=` because the sequence files contain their own start times.

id=*hex* Specify stream ID (lower 8 bits of packet startcode). The valid ranges are 0xC0-0xDF for audio, 0xE0-0xEF for video, and 0xF0-0xFF for data. Default allocation is the lowest unused ID for this stream kind. This option is ignored for CD, because the standards specify the correct value for each of the stream types.

DESCRIPTION

Combine elementary video, audio and data stream into a single MPEG 1 system stream. All of the elementary streams are assumed to be constant bit rate (CBR) and all I/O is from files [for now].

Timestamps (PTS and optionally DTS) are pre-appended to each presentation unit.

ABOUT STILL PICTURES...

Still pictures are a nice addition to CD-i and Video CD 2.0 formats (Note that). CD-i and Video CD have different limitations for still pictures summarized in the table below . Prior to version 2.0, the Video CD specification did not allow still pictures, and now they are only allowed in the segment play area, which is the CDROM track 1. Both allow mixing video, audio and still pictures.

	CD-i (`format=cdi`)	Video CD Segment Play (`format=vcdseg`)
NTSC picture size	maximum area of 221,184 pixels. For example 384x576	normal (352x240) and high (704x480) resolution
PAL picture size	same as NTSC	normal (352x88) and high (704x576) resolution
Number of still streams	16	two: one normal and one high resoltion stream
Data size	47,104 bytes	47,104 bytes for normal resolution; 229,376 bytes for high resolution

About the only other constraints on still pictures is that you must leave enough between pictures to load the data. You may be surprised to know that although quadruple speed cdrom drives are available, CD-i and Video CD players cannot take advantage of them, because the sector rate is defined to be 75 sectors per second! Each sector on the cdrom is 2352 bytes, but after you subtract the various overhead information, it can contain only 2294 bytes of pure video data. For example, if you have nothing but video on your cdrom, the mximum data rate is 2,294 X 75 = 172,050 bytes per second. If your still pictures are 25,000 bytes each, then you cannot display them faster than about 7 frames per second; less if you include audio; even less if you have more than one still picture stream.

STILL PICTURE FILE FORMAT

The format consist of three fields separated by white space on a single line. The # character introduces a comment; the remaining part of the line is ignored. The fields from left-to-right are timecode, low-res picture, high-res picture. The timecode is specified in hour:min:sec:-frame (the meaning of frame comes from the `tcfps` parameter). The second and third field specify file names that contain the still frames; the path may be relative or absolute (preceeded by a '/' character). The second field is required; the third field is also is optional. For example:

```
# still picture segment play list
00:00:00:00    pict1-low   pict1-high
00:00:05:14    pict2-low
00:00:10:00    pict3-low   pict3-high
00:00:15:05    pict4-low
```

Although Video CD and CD-i specify that pictures be encoded as variable bit-rate, certain picture rates, and so on, most of these are "fixed up" by `mpegsmux` during the multiplexing process. A few cannot, such as picture size and buffer size, because it affects the encoding of the entire sequence. `Mpegsmux` will tell you what it can and can't fix.

Note that although CD-i allows up to 16 still picture sequences, mpegsmux can only handle one at this time. You can have as many pictures in the still sequence, but there can only be one sequence.

ABOUT VIDEO SEQUENCES...

For a generic system stream (i.e., not CD-i and not Video CD), there is only one limitation: it must be encoded as a fixed bitrate stream. This is not a requirement of the MPEG standard, but it is a practical limitation for this software.

CD-i and Video CD 2.0 formats have a number of limitations for video sequences; some are the same, some slightly different. The important ones are summarized in the table below. There are a few other limitations, but these are taken care of by specifying `format=cdi`, `vcdseg` or `videocd`. Consult the standards documents for details.

	CD-i (`format=cdi`)	Video CD (`format=vcdseg` or `videocd`)
picture size and frame rate	Maximum rate of 9,900 macroblocks (16x16 pixels) per second. For example: 352x240 @ 30 Hz 352x240 @ 29.97 Hz 352x288 @ 25 Hz 352x288 @ 25 Hz 352x288 @ 23.976 Hz	352x240 @ 29.97 Hz 352x240 @ 23.976 Hz 352x288 @ 25 Hz
Bit rate	Max. of approx. 1,400,000 bits per second from disk; 5,000,000 bits/sec from memory.	Segment Play (track 1): 1,377,600 bits/sec. max. All other tracks: 1,151,930 bits/sec. max.
Buffer size	Max of 40,960 bytes	Same
Motion vectors	+/- 128 pixels max. (full-pixel) +/- 64 pixels max. (half-pixel)	Same
Number of streams	16 max.	1 max.

The buffer size and motion vectors are complex, and fall under the nomenclature of "Constrained Parameter Streams". For simplicity, NVR's video compressor includes support for creating such streams. Simply specify the `-cpf` flag to `vmpegcomp`, or by selecting CD-i or Video CD in the Compressor GUI.

ABOUT AUDIO SEQUENCES...

As with video, audio streams in a generic system stream have only one limitation: it must be encoded as a fixed bitrate audio stream. This is not a requirement of the MPEG standard, but it is a practical limitation for this software.

CD-i and Video CD 2.0 formats have a few limitations for audio sequences. The important ones are summarized in the table below. There are a few other limitations, but these are taken care of by specifying `format=cdi`, `vcdseg` or `videocd`. Consult the standards documents for details.

	CD-i (`format=cdi`)	Video CD Segment Play (`format=vcdseg`)	Video CD Audio/Video (`format=videocd`)
Audio coding layer	Layer I or Layer II	Layer I or Layer II	Layer II only
Bit rate	any	224 kBits/sec.	Mono: 64 kBits/sec. 96 kBits/sec. 192 kBits/sec. Stereo, etc.: 128 kBits/sec. 192 kBits/sec. 224 kBits/sec. 384 kBits/sec.
Sampling Frequency	44.1 kHz	Same	Same
Emphasis	None or 50/15 µsec.	Same	Same
Number of streams	32	1	1

ABOUT SEQUENCE LIST FILES...

A sequence list file is a way to put several sequences together as if they are one. This can be a little confusing because the MPEG System Stream specification provides a similar capability, but its not quite the same. A generic MPEG system stream may contain up to 32 audio streams and 16 video streams, each with a different stream id. The intent is for these streams to be played in parallel. In contrast, a sequence list file allows you to take several video or audio sequences and and encode them serially (one after the other) in a single video or audio stream. This is especially helpful if you encoded a movie in “chunks” which is sometimes the case with long movies because of the vast disk storage requirements.

A sequence list file simply allows you to list each chunk (or file), one file per line. In addition, if you want a pause between each sequence, you can specify a start time for each file. Video CD and CD-i include nformation about a system stream at the sector level and these support the concept of a pause or "chapter mark". You can cause a chapter mark by including an EOR string at the end of each line.

SEQUENCE LIST FILE FORMAT

The format consist of one field and two optional fields separated by white space on a single line. The fields from left-to-right are filename, timecode, EOR (end of record) marker. The timecode is specified in hour:min:sec:fraction. The # character introduces a comment and the remaining part of the line is ignored. The EOR marker is meaningful only for CDROM applications where an EOR bit may be set in the sector header.

For most applications, it is sufficient to simply list the files, and the effect is that all sequences will run together. If a timecode is provided, then that sequence is delayed until then. For example::

```
# video list
sequence1.vmpeg
sequence2.vmpeg
```

Another possiblility:

```
# timed audio list
sequence1.ampeg            00:00:00:00
sequence2.ampeg            00:00:45:00
```

SEE ALSO

player, recorder, compressor, vmpegcomp, ampegcomp

EXIT CODES

For the benefit of those doing shell programming with mpegsmux, the following exit codes are used by mpegsmux:

Exit value	Meaning
0	No error
1	Miscellaneous errors
2	Internal problem
3	A video or audio file is encoded incorrectly
4	Write failed because disk is full
5	Command line parameter error, or syntax error in sequence list or still files

COPYRIGHT

BUGS

No data streams are allowed on compact discs.

Only `videocd`, `vcdseg` and `cdi` formats can contain still pictures. Generic system streams cannot.

Although the MPEG standard allows it, variable rate video and audio sequeces are not allowed in generic system streams. CD-i and Video CD standards do not allow variable rate streams.