NAME

mpg123 - play audio MPEG 1.0/2.0/2.5 stream (layers 1, 2 and 3)

SYNOPSIS

mpg123 [*options*] *file-or-URL*...

DESCRIPTION

mpg123 reads one or more *file*s (or standard input if ‘‘**-**’’ is specified) or *URL*s and plays them on the audio device (default) or outputs them to stdout. *file*/*URL* is assumed to be an MPEG audio bit stream.

OPERANDS

The following operands are supported:

file(s) The path name(s) of one or more input files. They must be valid MPEG-1.0/2.0/2.5 audio layer 1, 2 or 3 bit streams. If a dash ‘‘**-**’’ is specified, MPEG data will be read from the standard input. Furthermore, any name starting with ‘‘http://’’ or ‘‘https://’’ is recognized as *URL* (see next section), while a leading ‘‘file://’’ is being stripped for normal local file access, for consistency (since mpg123 1.30.1).

OPTIONS

mpg123 options may be either the traditional POSIX one letter options, or the GNU style long options. POSIX style options start with a single ‘‘**-**’’, while GNU long options start with ‘‘**--**’’. Option arguments (if needed) follow separated by whitespace (not ‘‘=’’). Note that some options can be absent from your installation when disabled in the build process.

INPUT OPTIONS

-k *num*, **--skip** *num*

Skip first *num* frames. By default the decoding starts at the first frame.

-n *num*, **--frames** *num*

Decode only *num* frames. By default the complete stream is decoded.

--fuzzy

Enable fuzzy seeks (guessing byte offsets or using approximate seek points from Xing TOC). Without that, seeks need a first scan through the file before they can jump at positions. You can decide here: sample-accurate operation with gapless features or faster (fuzzy) seeking.

-y, **--no-resync**

Do NOT try to resync and continue decoding if an error occurs in the input file. Normally, **mpg123** tries to keep the playback alive at all costs, including skipping invalid material and searching new header when something goes wrong. With this switch you can make it bail out on data errors (and perhaps spare your ears a bad time). Note that this switch has been renamed from --resync. The old name still works, but is not advertised or recommended to use (subject to removal in future).

-F, **--no-frankenstein**

Disable support for Frankenstein streams. Normally, mpg123 stays true to the concept of MPEG audio being just a concatenation of MPEG frames. It will continue decoding even if the type of MPEG frames varies wildly. With this switch, it will only decode the input as long as it does not change its character (from layer I to layer III, changing sampling rate, from mono to stereo), silently assuming end of stream on such occasion. The switch also stops decoding of compatible MPEG frames if there was an Info frame (Xing header, Lame tag) that contained a length of the track in MPEG frames. This comes a bit closer to the notion of a MP3 file as a defined collection of MPEG frames that belong together, but gets rid of the flexibility that can be fun at times but mostly is hell for the programmer of the parser and decoder ...

--network *backend*

Select network backend (helper program), choices are usually auto, wget, and curl. Auto means to try the first available backend.

--resync-limit *bytes*

Set number of bytes to search for valid MPEG data once lost in stream; <0 means search whole stream. If you know there are huge chunks of invalid data in your files... here is your hammer. Note: Only since version 1.14 this also increases the amount of junk skipped on beginning.

-u *auth*, **--auth** *auth*

HTTP authentication to use when receiving files via HTTP. The format used is user:password. Mpg123 will clear this quickly, but it may still appear in sight of other users or even just in your shell history. You may seek alternative ways to specify that to your network backend.

--auth-file *authfile*

Provide the authentication info via given file instead of command line directly.

--ignore-mime

Ignore MIME types given by HTTP server. If you know better and want mpg123 to decode something the server thinks is image/png, then just do it.

--no-icy-meta

Do not accept ICY meta data.

--streamdump *filename*

Dump a copy of the input data (as read by libmpg123) to the given file. This enables you to store a web stream to disk while playing, or just create a concatenation of the local files you play for ... why not?

--icy-interval *bytes*

This setting enables you to play a stream dump containing ICY metadata at the given interval in bytes (the value of the icy-metaint HTTP response header). Without it, such a stream will play, but will cause regular decoding glitches with resync.

--no-seekbuffer

Disable the default micro-buffering of non-seekable streams that gives the parser a safer footing.

-@ *file*, **--list** *file*

Read filenames and/or URLs of MPEG audio streams from the specified *file* in addition to the ones specified on the command line (if any). Note that *file* can be either an ordinary file, a dash ‘‘-’’ to indicate that a list of filenames/URLs is to be read from the standard input, or an URL pointing to a an appropriate list file. Note: only one **-@** option can be used (if more than one is specified, only the last one will be recognized). Furthermore, for HTTP resources, the MIME type information will be used to re-open an actual MPEG stream as such instead of treating it as playlist file. So you could just always use **-@** for web resources without bothering if it is a playlist or already the resolved stream address.

-l *n*, **--listentry** *n*

Of the playlist, play specified entry only. *n* is the number of entry starting at 1. A value of 0 is the default and means playing the whole list, a negative value means showing of the list of titles with their numbers...

--continue

Enable playlist continuation mode. This changes frame skipping to apply only to the first track and also continues to play following tracks in playlist after the selected one. Also, the option to play a number of frames only applies to the whole playlist. Basically, this tries to treat the playlist more like one big stream (like, an audio book). The current track number in list (1-based) and frame number (0-based) are printed at exit (useful if you interrupted playback and want to continue later). Note that the continuation info is printed to standard output unless the switch for piping audio data to standard out is used. Also, it really makes sense to work with actual playlist files instead of lists of file names as arguments, to keep track positions consistent.

--loop *times*

for looping track(s) a certain number of times, < 0 means infinite loop (not with --random!).

--keep-open
For remote control mode: Keep loaded file open after reaching end.

--timeout *seconds*
Timeout in (integer) seconds before declaring a stream dead (if <= 0, wait forever).

-z, **--shuffle**
Shuffle play. Randomly shuffles the order of files specified on the command line, or in the list file.

-Z, **--random**
Continuous random play. Keeps picking a random file from the command line or the play list. Unlike shuffle play above, random play never ends, and plays individual songs more than once.

-i, --index
Index / scan through the track before playback. This fills the index table for seeking (if enabled in libmpg123) and may make the operating system cache the file contents for smoother operating on playback.

--index-size *size*
Set the number of entries in the seek frame index table.

--preframes *num*
Set the number of frames to be read as lead-in before a seeked-to position. This serves to fill the layer 3 bit reservoir, which is needed to faithfully reproduce a certain sample at a certain position. Note that for layer 3, a minimum of 1 is enforced (because of frame overlap), and for layer 1 and 2, this is limited to 2 (no bit reservoir in that case, but engine spin-up anyway).

OUTPUT and PROCESSING OPTIONS

-o *module*, **--output** *module*
Select audio output module. You can provide a comma-separated list to use the first one that works. Also see **-a**.

--list-modules
List the available modules.

--list-devices
List the available output devices for given output module. If there is no functionality to list devices in the chosen module, an error will be printed and mpg123 will exit with a non-zero code.

-a *dev*, **--audiodevice** *dev*
Specify the audio device to use. The default as well as the possible values depend on the active output. For the JACK output, a comma-separated list of ports to connect to (for each channel) can be specified.

-s, **--stdout**
The decoded audio samples are written to standard output, instead of playing them through the audio device. This option must be used if your audio hardware is not supported by **mpg123**. The output format per default is raw (headerless) linear PCM audio data, 16 bit, stereo, host byte order (you can force mono or 8bit).

-O *file*, **--outfile**
Write raw output into a file (instead of simply redirecting standard output to a file with the shell).

-w *file*, **--wav**
Write output as WAV file. This will cause the MPEG stream to be decoded and saved as file *file* , or standard output if **-** is used as file name. You can also use *--au* and *--cdr* for AU and CDR format, respectively. Note that WAV/AU writing to non-seekable files, or redirected stdout, needs some thought. Since 1.16.0, the logic changed to writing the header with the first actual data. This avoids spurious WAV headers in a pipe, for example. The result of decoding nothing to WAV/AU is a file consisting just of the header when it is seekable and really nothing when not (not even a header). Correctly writing data with prophetic headers to stdout is no easy business.

--au *file*
Does not play the MPEG file but writes it to *file* in SUN audio format. If **-** is used as the filename, the AU file is written to stdout. See paragraph about WAV writing for header fun with non-seekable streams.

--cdr *file*
Does not play the MPEG file but writes it to *file* as a CDR file. If **-** is used as the filename, the CDR file is written to stdout.

--reopen
Forces reopen of the audiodevice after ever song

--cpu *decoder-type*
Selects a certain decoder (optimized for specific CPU), for example i586 or MMX. The list of available decoders can vary; depending on the build and what your CPU supports. This option is only available when the build actually includes several optimized decoders.

--test-cpu
Tests your CPU and prints a list of possible choices for --cpu.

--list-cpu
Lists all available decoder choices, regardless of support by your CPU.

-g *gain*, **--gain** *gain*
[DEPRECATED] Set audio hardware output gain (default: don't change). The unit of the gain value is hardware and output module dependent. (This parameter is only provided for backwards compatibility and may be removed in the future without prior notice. Use the audio player for playing and a mixer app for mixing, UNIX style!)

-f *factor*, **--scale** *factor*
Change scale factor (default: 32768).

--rva-mix, --rva-radio
Enable RVA (relative volume adjustment) using the values stored for ReplayGain radio mode / mix mode with all tracks roughly equal loudness. The first valid information found in ID3V2 Tags (Comment named RVA or the RVA2 frame) or ReplayGain header in Lame/Info Tag is used.

--rva-album, --rva-audiophile
Enable RVA (relative volume adjustment) using the values stored for ReplayGain audiophile mode / album mode with usually the effect of adjusting album loudness but keeping relative loudness inside album. The first valid information found in ID3V2 Tags (Comment named RVA_ALBUM or the RVA2 frame) or ReplayGain header in Lame/Info Tag is used.

-0, **--single0**; **-1**, **--single1**
Decode only channel 0 (left) or channel 1 (right), respectively. These options are available for stereo MPEG streams only.

-m, **--mono**, **--mix**, **--singlemix**
Mix both channels / decode mono. It takes less CPU time than full stereo decoding.

--stereo
Force stereo output

-r *rate*, **--rate** *rate*
Set sample rate (default: automatic). You may want to change this if you need a constant bitrate independent of the mpeg stream rate. mpg123 automagically converts the rate. You should then combine this with --stereo or --mono.

--resample *method*

Set resampling method to employ if forcing an output rate. Choices (case-insensitive) are NtoM, dirty, and fine. The fine resampler is the default. It employs libsyn123's low-latency fairly efficient resampler to postprocess the output from libmpg123 instead of the fast but very crude NtoM decoder (drop sample method) that mpg123 offers since decades. If you are really low on CPU time, choose NtoM, as the resampler usually needs more time than the MPEG decoder itself. The mpg123 program is smart enough to combine the 2to1 or 4to1 downsampling modes with the postprocessing for extreme downsampling.

-2, **--2to1**; **-4**, **--4to1**

Performs a downsampling of ratio 2:1 (22 kHz from 44.1 kHz) or 4:1 (11 kHz) on the output stream, respectively. Saves some CPU cycles, but of course throws away the high frequencies, as the decoder does not bother producing them.

--pitch *value*

Set a pitch change (speedup/down, 0 is neutral; 0.05 is 5% speedup). When not enforcing an output rate, this changes the output sampling rate, so it only works in the range your audio system/hardware supports. When you combine this with a fixed output rate, it modifies a software resampling ratio instead.

--8bit

Forces 8bit output

--float

Forces f32 encoding

-e *enc*, **--encoding** *enc*

Choose output sample encoding. Possible values look like f32 (32-bit floating point), s32 (32-bit signed integer), u32 (32-bit unsigned integer) and the variants with different numbers of bits (s24, u24, s16, u16, s8, u8) and also special variants like ulaw and alaw 8-bit. See the output of mpg123's longhelp for actually available encodings.

-d *n*, **--doublespeed** *n*

Only play every *n*'th frame. This will cause the MPEG stream to be played *n* times faster, which can be used for special effects. Can also be combined with the **--halfspeed** option to play 3 out of 4 frames etc. Don't expect great sound quality when using this option.

-h *n*, **--halfspeed** *n*

Play each frame *n* times. This will cause the MPEG stream to be played at *1*/*n*'th speed (n times slower), which can be used for special effects. Can also be combined with the **--doublespeed** option to double every third frame or things like that. Don’t expect great sound quality when using this option.

-E *file*, **--equalizer**

Enables equalization, taken from *file*. The file needs to contain 32 lines of data, additional comment lines may be prefixed with *#*. Each data line consists of two floating-point entries, separated by whitespace. They specify the multipliers for left and right channel of a certain frequency band, respectively. The first line corresponds to the lowest, the 32nd to the highest frequency band. Note that you can control the equalizer interactively with the generic control interface. Also note that these are the 32 bands of the MPEG codec, not spaced like you would see for a usual graphic equalizer. The upside is that there is zero computational cost in addition to decoding. The downside is that you roughly have bass in band 0, (upper) mids in band 1, treble in all others.

--gapless

Enable code that cuts (junk) samples at beginning and end of tracks, enabling gapless transitions between MPEG files when encoder padding and codec delays would prevent it. This is enabled per default beginning with mpg123 version 1.0.0 .

--no-gapless

Disable the gapless code. That gives you MP3 decodings that include encoder delay and padding plus mpg123’s decoder delay.

--no-infoframe

Do not parse the Xing/Lame/VBR/Info frame, decode it instead just like a stupid old MP3 hardware player. This implies disabling of gapless playback as the necessary information is in said metadata frame.

-D *n*, **--delay** *n*

Insert a delay of *n* seconds before each track.

-o h, **--headphones**

Direct audio output to the headphone connector (some hardware only; AIX, HP, SUN).

-o s, **--speaker**

Direct audio output to the speaker (some hardware only; AIX, HP, SUN).

-o l, **--lineout**

Direct audio output to the line-out connector (some hardware only; AIX, HP, SUN).

-b *size*, **--buffer** *size*

Use an audio output buffer of *size* Kbytes. This is useful to bypass short periods of heavy system activity, which would normally cause the audio output to be interrupted. You should specify a buffer size of at least 1024 (i.e. 1 Mb, which equals about 6 seconds of audio data) or more; less than about 300 does not make much sense. The default is 0, which turns buffering off.

--preload *fraction*

Wait for the buffer to be filled to *fraction* before starting playback (fraction between 0 and 1). You can tune this prebuffering to either get faster sound to your ears or safer uninterrupted web radio. Default is 0.2 (wait for 20 % of buffer to be full, changed from 1 in version 1.23).

--devbuffer *seconds*

Set device buffer in seconds; <= 0 means default value. This is the small buffer between the application and the audio backend, possibly directly related to hardware buffers.

--smooth

Keep buffer over track boundaries **--** meaning, do not empty the buffer between tracks for possibly some added smoothness.

MISC OPTIONS

-t, **--test**

Test mode. The audio stream is decoded, but no output occurs.

-c, **--check**

Check for filter range violations (clipping), and report them for each frame if any occur.

-v, **--verbose**

Increase the verbosity level. For example, displays the frame numbers during decoding.

-q, **--quiet**

Quiet. Suppress diagnostic messages.

-C, **--control**

Enable terminal control keys. This is enabled automatically if a terminal is detected. By default use ’s’ or the space bar to stop/restart (pause, unpause) playback, ’f’ to jump forward to the next song, ’b’ to jump back to the beginning of the song, ’,’ to rewind, ’.’ to fast forward, and ’q’ to quit. Type ’h’ for a full list of available controls. The A-B loop feature with key ’o’ changes the preset loop interval to the interval between two presses of ’o’, the third press (or ’p’) ending the looped playback. The key ’p’ will use the updated loop interval after that.

--no-control
Disable terminal control even if terminal is detected.

--title
In an xterm, rxvt, screen, iris-ansi (compatible, TERM environment variable is examined), change the window's title to the name of song currently playing.

--pauseloop *seconds*
Set the length of the loop interval in terminal control fixed looping mode, away from the default of 0.5 seconds, as a floating point number. This value can be overwritten at runtime using the A-B loop feature.

--name *name*
Set the name of this instance, possibly used in various places. This sets the client name for JACK output.

--long-tag
Display ID3 tag info always in long format with one line per item (artist, title, ...)

--utf8
Regardless of environment, print metadata in UTF-8 (otherwise, when not using UTF-8 locale, you'll get ASCII stripdown).

-R, **--remote**
Activate generic control interface. **mpg123** will then read and execute commands from stdin. Basic usage is ‘‘load <filename> ’’ to play some file and the obvious ‘‘pause’’, ‘‘command. ‘‘jump <frame>’’ will jump/seek to a given point (MPEG frame number). Issue ‘‘help’’ to get a full list of commands and syntax.

--remote-err
Print responses for generic control mode to standard error, not standard out. This is automatically triggered when using **-s**.

--fifo *path*
Create a fifo / named pipe on the given path and use that for reading commands instead of standard input.

--aggressive
Tries to get higher priority

-T, **--realtime**

Tries to gain realtime priority. This option usually requires root privileges to have any effect.

-?, **--help**

Shows short usage instructions.

--longhelp

Shows long usage instructions.

--version

Print the version string.

--libversion

Print version information on the mpg123 libraries being used (libmpg123, libout123, libsyn123).

HTTP SUPPORT

In addition to reading MPEG audio streams from ordinary files and from the standard input, **mpg123** supports retrieval of MPEG audio streams or playlists via the HTTP protocol, which is used in the World Wide Web (WWW). Such files are specified using a so-called URL, which starts with http:// or https://. When a file with that prefix is encountered, **mpg123** since 1.30.0 will by default call an external helper program (either **wget**(1) or **curl**(1), see the **--network** option) to retrieve the resource. You can configure access via a proxy server using the standard environment variables those programs support. The **--proxy** option that **mpg123** before 1.30.0 used for its internal network code is gone in the default build now and will probably disappear for good with 1.31.1.

Note that, in order to play MPEG audio files from a WWW server, it is necessary that the connection to that server is fast enough. For example, a 128 kbit/s MPEG file requires the network connection to be at least 128 kbit/s (16 kbyte/s) plus protocol overhead. If you suffer from short network outages, you should try the **-b** option (buffer) to bypass such outages. If your network connection is generally not fast enough to retrieve MPEG audio files in realtime, you can first download the files to your local harddisk (e.g. using **wget**(1)) and then play them from there.

Streams with embedded ICY metadata are supported, the interval being communicated via HTTP headers or **--icy-interval**.

INTERRUPT

When in terminal control mode, you can quit via pressing the q key, while any time you can abort **mpg123** by pressing Ctrl-C. If not in terminal control mode, this will skip to the next file (if any). If you want to abort playing immediately in that case, press Ctrl-C twice in short succession (within about one second).

Note that the result of quitting **mpg123** pressing Ctrl-C might not be audible immediately, due to audio data buffering in the audio device. This delay is system dependent, but it is usually not more than one or two seconds.

PLAYBACK STATUS LINE

In verbose mode, mpg123 updates a line with various information centering around the current playback position. On any decent terminal, the line also works as a progress bar in the current file by reversing video for a fraction of the line according to the current position. An example for a full line is this:

> 0291+0955 00:01.68+00:28.22 [00:05.30] mix 100=085 192 kb/s 576 B acc 18 clip p+0.014

The information consists of, in order:

> single-character playback state (‘‘>’’ for playing, ‘‘=’’ for pausing/looping, ‘‘_’’ for stopped)

0291+0955
current frame offset and number of remaining frames after the plus sign

00:01.68+00:28.22
current position from and remaining time in human terms (hours, minutes, seconds)

[00:05.30]
fill of the output buffer in terms of playback time, if the buffer is enabled

mix
selected RVA mode (possible values: mix, alb (album), and --- (neutral, off))

100=085
set volume and the RVA-modified effective volume after the equal sign

192 kb/s
current bitrate

576 B
size of current frame in bytes

acc if positions are accurate, possible values are ‘‘acc’’ for accurate positions or ‘‘fuz’’ for fuzzy (with guessed byte offsets using mean frame size)

18 clip

amount of clipped samples, non-zero only if decoder reports that (generic does, some optimized ones not)

p+0.014

pitch change (increased/decreased playback sampling rate on user request)

NOTES

MPEG audio decoding requires a good deal of CPU performance, especially layer-3. To decode it in realtime, you should have at least an i486DX4, Pentium, Alpha, SuperSparc or equivalent processor. You can also use the **-m** option to decode mono only, which reduces the CPU load somewhat for layer-3 streams. See also the **-2** and **-4** options.

If everything else fails, have mpg123 decode to a file and then use an appropriate utility to play that file with less CPU load. Most probably you can configure mpg123 to produce a format suitable for your audio device (see above about encodings and sampling rates).

If your system is generally fast enough to decode in realtime, but there are sometimes periods of heavy system load (such as cronjobs, users logging in remotely, starting of ‘‘big’’ programs etc.) causing the audio output to be interrupted, then you should use the **-b** option to use a buffer of reasonable size (at least 1000 Kbytes).

EXIT CODE

Up to version 1.25.x, mpg123 always returned exit code 0 also for complete junk on the input side. Fatal errors were only considered for output. With version 1.26.0, this changed to the behaviour described below.

When not using the remote control interface (which returns input errors as text messages), the process exit code is zero (success) only if all tracks in a playlist had at least one frame parsed, even if it did not decode cleanly, or are empty, MPEG-wise (perhaps only metadata, or really an empty file). When you decode nothing, nothing is the result and that is fine. When a track later aborts because of parser errors or breakdown of the network communication, this is treated as end of a track, but does not make the process as such fail. One really bad (or non-existing) stream in the playlist results in a non-zero error code, consistent with other UNIX tools.

An error in audio output results in the process ending with a non-zero exit code immediately, regardless of how much data has been successfully played before. The forgiveness is only on the input side.

BUGS

Mostly MPEG-1 layer 2 and 3 are tested in real life. Please report any issues and provide test files to help fixing them.

No CRC error checking is performed. But the decoder is built and tested to behave nicely with damaged streams. Mostly, damaged frames will just be silent.

Some platforms lack audio hardware support; you may be able to use the **-s** switch to feed the decoded data to a program that can play it on your audio device.

AUTHORS

Maintainer:

Thomas Orgis <maintainer@mpg123.org>, <thomas@orgis.org>

Original Creator:

Michael Hipp

Uses code or ideas from various people, see the AUTHORS file accompanying the source code.

LICENSE

mpg123 is licensed under the GNU Lesser/Library General Public License, LGPL, version 2.1 .

WEBSITE

http://www.mpg123.org
http://sourceforge.net/projects/mpg123