NAME

gsm_option -- customizing the GSM 06.10 implementation

SYNOPSIS

```
#include "gsm.h"

int gsm_option(handle, option, valueP);
gsm handle;
int option;
int * valueP;
```

DESCRIPTION

The gsm library is an implementation of the final draft GSM 06.10 standard for full-rate speech transcoding, a lossy speech compression algorithm.

The gsm_option() function can be used to set and query various options or flags that are not needed for regular GSM 06.10 encoding or decoding, but might be of interest in special cases.

The second argument to gsm_option specifies what parameter should be changed or queried. The third argument is either a null pointer, in which case the current value of that parameter is returned; or it is a pointer to an integer containing the value you want to set, in which case the previous value will be returned.

The following options are defined:

GSM_OPT_VERBOSE Verbosity level.
This option is only supported if the library was compiled with debugging turned on, and may be used by developers of compression algorithms to aid debugging.
The verbosity level can be changed at any time during encoding or decoding.

GSM_OPT_FAST Faster compression algorithm.
This implementation offers a not strictly standard-compliant, but faster compression algorithm that is compatible with the regular method and does not noticably degrade audio quality.
The value passed to

```
gsm_option(handle, GSM_OPT_FAST, & value)
```

functions as a boolean flag; if it is zero, the regular algorithm will be used, if not, the faster version will be used.
The availability of this option depends on the hardware used; if it is not available, gsm_option will return -1 on an attempt to set or query it.

This option can be set any time during encoding or decoding.

GSM_OPT_LTP_CUT Enable, disable, or query the LTP cut-off optimization.
During encoding, the search for the long-term correlation lag forms the bottleneck of the algorithm. The ltp-cut option enables an approximation that disregards most of the samples for purposes of finding that correlation, and hence speeds up the encoding at a noticable loss in quality.
The value passed to

gsm_option(handle, GSM_OPT_LTP_CUT, & value)

turns the optimization on if nonzero, and off if zero.
This option can be set any time during encoding or decoding; it will only affect the encoding pass, not the decoding.

GSM_OPT_WAV49 WAV-style byte ordering.
A WAV file of type #49 contains GSM 06.10-encoded frames. Unfortunately, the framing and code ordering of the WAV version are incompatible with the native ones of this GSM 06.10 library. The GSM_OPT_WAV49 option turns on a different packing algorithm that produces alternating frames of 32 and 33 bytes (or makes it consume alternating frames of 33 and 32 bytes, note the opposite order of the two numbers) which, when concatenated, can be used in the body of a WAV #49 frame. It is up to the user program to write a WAV header, if any; neither the library itself nor the toast program produce complete WAV files.
The value passed to

gsm_option(handle, GSM_OPT_WAV49, & value)

functions as a boolean flag; if it is zero, the library's native framing algorithm will be used, if nonzero, WAV-type packing is in effect.
This option should be used before any frames are encoded. Whether or not it is supported at all depends on a compile-time switch, WAV49. Both option and compile time switch are new to the library as of patchlevel 9, and are considerably less tested than the well-worn rest of the it.
Thanks to Jeff Chilton for the detective work and first free implementation of this version of the GSM 06.10 encoding.

GSM_OPT_FRAME_CHAIN Query or set the chaining byte.
Between the two frames of a WAV-style encoding, the GSM 06.10 library must keep track of one half-byte that is technically part of the first frame, but will be written as the first four bits of the second. This half-byte are the lowest four bits of the value returned by, and optionally set by,

gsm_option(handle, GSM_OPT_FRAME_CHAIN, & value)

This option can be queried and set at any time.

GSM_OPT_FRAME_INDEX Query or set the current frame's index in a format's alternating list of frames.

The WAV #49 framing uses two alternating types of frames. Which type the next GSM-coded frame belongs to can be queried, or, when decoding, announced, using

gsm_option(handle, GSM_OPT_FRAME_INDEX, & value)

For WAV-style framing, the value should be 0 or 1; the first frame of an encoding has an index of 0. At library initialization, the index is set to zero.

The frame index can be queried and set at any time. Used in combination with the *GSM_OPT_FRAME_CHAIN*, option, it can be used to position on arbitrary GSM frames within a format like WAV #49 (not accounting for the lost internal GSM state).

RETURN VALUE

gsm_option() returns -1 if an option is not supported, the previous value of the option otherwise.

BUGS

Please direct bug reports to jutta@pobox.com and cabo@tzi.org.

SEE ALSO

toast(1), gsm(3), gsm_explode(3), gsm_print(3)