CELT

0.7.0

Generated by Doxygen 1.6.1

Mon Oct 26 10:08:02 2009

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Module Documentation

4.1 Encoding and decoding

Functions

- EXPORT CELTMode ∗ celt_mode_create (celt_int32 Fs, int frame_size, int ∗error)
- EXPORT void celt_mode_destroy (CELTMode ∗mode)
- EXPORT int celt_mode_info (const CELTMode ∗mode, int request, celt_int32 ∗value)
- EXPORT CELTEncoder ∗ celt_encoder_create (const CELTMode ∗mode, int channels, int ∗error)
- EXPORT void celt_encoder_destroy (CELTEncoder ∗st)
- EXPORT int celt_encode_float (CELTEncoder ∗st, const float ∗pcm, float ∗optional_synthesis, unsigned char ∗compressed, int nbCompressedBytes)
- EXPORT int celt_encode (CELTEncoder ∗st, const celt_int16 ∗pcm, celt_int16 ∗optional_synthesis, unsigned char ∗compressed, int nbCompressedBytes)
- EXPORT int celt_encoder_ctl (CELTEncoder ∗st, int request,...)
- EXPORT CELTDecoder ∗ celt_decoder_create (const CELTMode ∗mode, int channels, int ∗error)
- EXPORT void celt_decoder_destroy (CELTDecoder ∗st)
- EXPORT int celt_decode_float (CELTDecoder ∗st, const unsigned char ∗data, int len, float ∗pcm)
- EXPORT int celt_decode (CELTDecoder ∗st, const unsigned char ∗data, int len, celt_int16 ∗pcm)
- EXPORT int celt_decoder_ctl (CELTDecoder ∗st, int request,...)
- EXPORT const char ∗ celt_strerror (int error)

4.1.1 Detailed Description

4.1.2 Function Documentation

4.1.2.1 EXPORT int celt_decode (CELTDecoder ∗ *st*, const unsigned char ∗ *data*, int *len*, celt_int16 ∗ *pcm*)

Decodes a frame of audio.

Parameters:

st Decoder state

data Compressed data produced by an encoder

len Number of bytes to read from "data". This MUST be exactly the number of bytes returned by the encoder. Using a larger value WILL NOT WORK.

pcm One frame (frame_size samples per channel) of decoded PCM will be returned here in 16-bit PCM format (native endian).

Returns:

Error code.

4.1.2.2 EXPORT int celt_decode_float (CELTDecoder ∗ *st*, const unsigned char ∗ *data*, int *len*, float ∗ *pcm*)

Decodes a frame of audio.

Parameters:

st Decoder state

data Compressed data produced by an encoder

len Number of bytes to read from "data". This MUST be exactly the number of bytes returned by the encoder. Using a larger value WILL NOT WORK.

pcm One frame (frame_size samples per channel) of decoded PCM will be returned here in float format.

Returns:

Error code.

4.1.2.3 EXPORT CELTDecoder∗ celt_decoder_create (const CELTMode ∗ *mode*, int *channels*, int ∗ *error*)

Creates a new decoder state. Each stream needs its own decoder state (can't be shared across simultaneous streams).

Parameters:

mode Contains all the information about the characteristics of the stream (must be the same characteristics as used for the encoder)

channels Number of channels

error Returns an error code

Returns:

Newly created decoder state.

4.1.2.4 EXPORT int celt_decoder_ctl (CELTDecoder ∗ *st*, int *request*, ...)

Query and set decoder parameters

Parameters:

st Decoder state

request Parameter to change or query

value Pointer to a 32-bit int value

Returns:

Error code

4.1.2.5 EXPORT void celt_decoder_destroy (CELTDecoder ∗ *st*)

Destroys a a decoder state.

Parameters:

st Decoder state to be destroyed

4.1.2.6 EXPORT int celt_encode (CELTEncoder ∗ *st*, const celt_int16 ∗ *pcm*, celt_int16 ∗ *optional_synthesis*, unsigned char ∗ *compressed*, int *nbCompressedBytes*)

Encodes a frame of audio.

Parameters:

st Encoder state

pcm PCM audio in signed 16-bit format (native endian). There must be exactly frame_size samples per channel.

optional_synthesis If not NULL, the encoder copies the audio signal that the decoder would decode. It is the same as calling the decoder on the compressed data, just faster. This may alias pcm.

compressed The compressed data is written here. This may not alias pcm or optional_synthesis.

nbCompressedBytes Maximum number of bytes to use for compressing the frame (can change from one frame to another)

Returns:

Number of bytes written to "compressed". Will be the same as "nbCompressedBytes" unless the stream is VBR and will never be larger. If negative, an error has occurred (see error codes). It is IMPORTANT that the length returned be somehow transmitted to the decoder. Otherwise, no decoding is possible.

4.1.2.7 EXPORT int celt_encode_float (CELTEncoder ∗ *st*, const float ∗ *pcm*, float ∗ *optional_synthesis*, unsigned char ∗ *compressed*, int *nbCompressedBytes*)

Encodes a frame of audio.

Parameters:

st Encoder state

pcm PCM audio in float format, with a normal range of ±1.0. Samples with a range beyond ±1.0 are supported but will be clipped by decoders using the integer API and should only be used if it is known that the far end supports extended dynmaic range. There must be exactly frame_size samples per channel.

optional_synthesis If not NULL, the encoder copies the audio signal that the decoder would decode. It is the same as calling the decoder on the compressed data, just faster. This may alias pcm.

compressed The compressed data is written here. This may not alias pcm or optional_synthesis.

nbCompressedBytes Maximum number of bytes to use for compressing the frame (can change from one frame to another)

Returns:

Number of bytes written to "compressed". Will be the same as "nbCompressedBytes" unless the stream is VBR and will never be larger. If negative, an error has occurred (see error codes). It is IMPORTANT that the length returned be somehow transmitted to the decoder. Otherwise, no decoding is possible.

4.1.2.8 EXPORT CELTEncoder* celt_encoder_create (const CELTMode * *mode*, int *channels*, int * *error*)

Creates a new encoder state. Each stream needs its own encoder state (can't be shared across simultaneous streams).

Parameters:

mode Contains all the information about the characteristics of the stream (must be the same characteristics as used for the decoder)

channels Number of channels

error Returns an error code

Returns:

Newly created encoder state.

4.1.2.9 EXPORT int celt_encoder_ctl (CELTEncoder * *st*, int *request*, ...)

Query and set encoder parameters

Parameters:

st Encoder state

request Parameter to change or query

value Pointer to a 32-bit int value

Returns:

Error code

4.1.2.10 EXPORT void celt_encoder_destroy (CELTEncoder * *st*)

Destroys a an encoder state.

Parameters:

st Encoder state to be destroyed

4.1.2.11 EXPORT CELTMode∗ celt_mode_create (celt_int32 *Fs*, int *frame_size*, int ∗ *error*)

Creates a new mode struct. This will be passed to an encoder or decoder. The mode MUST NOT BE DESTROYED until the encoders and decoders that use it are destroyed as well.

Parameters:

Fs Sampling rate (32000 to 96000 Hz)

frame_size Number of samples (per channel) to encode in each packet (even values; 64 - 512)

error Returned error code (if NULL, no error will be returned)

Returns:

A newly created mode

4.1.2.12 EXPORT void celt_mode_destroy (CELTMode ∗ *mode*)

Destroys a mode struct. Only call this after all encoders and decoders using this mode are destroyed as well.

Parameters:

mode Mode to be destroyed

4.1.2.13 EXPORT int celt_mode_info (const CELTMode ∗ *mode*, int *request*, celt_int32 ∗ *value*)

Query information from a mode

4.1.2.14 EXPORT const char∗ celt_strerror (int *error*)

Returns the English string that corresponds to an error code

Parameters:

error Error code (negative for an error, 0 for success

Returns:

Constant string (must NOT be freed)

Chapter 5

Data Structure Documentation

5.1 CELTHeader Struct Reference

Header data.

```
#include <celt_header.h>
```

Data Fields

- char codec_id [8]
- char codec_version [20]
- celt_int32 version_id
- celt_int32 header_size
- celt_int32 sample_rate
- celt_int32 nb_channels
- celt_int32 frame_size
- celt_int32 overlap
- celt_int32 bytes_per_packet
- celt_int32 extra_headers

5.1.1 Detailed Description

Header data. Header data to be used for Ogg files (or possibly other encapsulation)

Definition at line 46 of file celt_header.h.

5.1.2 Field Documentation

5.1.2.1 celt_int32 CELTHeader::bytes_per_packet

Number of bytes per compressed packet (0 if unknown)

Definition at line 55 of file celt_header.h.

5.1.2.2 char CELTHeader::codec_id[8]

MUST be "CELT " (four spaces)

Definition at line 47 of file celt_header.h.

5.1.2.3 char CELTHeader::codec_version[20]

Version used (as string)

Definition at line 48 of file celt_header.h.

5.1.2.4 celt_int32 CELTHeader::extra_headers

Number of additional headers that follow this header

Definition at line 56 of file celt_header.h.

5.1.2.5 celt_int32 CELTHeader::frame_size

Samples per frame (per channel)

Definition at line 53 of file celt_header.h.

5.1.2.6 celt_int32 CELTHeader::header_size

Size of this header

Definition at line 50 of file celt_header.h.

5.1.2.7 celt_int32 CELTHeader::nb_channels

Number of channels

Definition at line 52 of file celt_header.h.

5.1.2.8 celt_int32 CELTHeader::overlap

Overlapping samples (per channel)

Definition at line 54 of file celt_header.h.

5.1.2.9 celt_int32 CELTHeader::sample_rate

Sampling rate of the original audio

Definition at line 51 of file celt_header.h.

5.1.2.10 celt_int32 CELTHeader::version_id

Version id (negative for until stream is frozen)

Definition at line 49 of file celt_header.h.

The documentation for this struct was generated from the following file:

- libcelt/celt_header.h

Chapter 6

File Documentation

6.1 libcelt/celt.h File Reference

Contains all the functions for encoding and decoding audio. `#include "celt_types.h"`

Defines

- #define CELT_OK 0
- #define CELT_BAD_ARG -1
- #define CELT_INVALID_MODE -2
- #define CELT_INTERNAL_ERROR -3
- #define CELT_CORRUPTED_DATA -4
- #define CELT_UNIMPLEMENTED -5
- #define CELT_INVALID_STATE -6
- #define CELT_ALLOC_FAIL -7
- #define CELT_GET_MODE(x) CELT_GET_MODE_REQUEST, _celt_check_mode_ptr_ptr(x)
- #define CELT_SET_COMPLEXITY(x) CELT_SET_COMPLEXITY_REQUEST, _celt_check_int(x)
- #define CELT_SET_PREDICTION(x) CELT_SET_PREDICTION_REQUEST, _celt_check_int(x)
- #define CELT_SET_VBR_RATE(x) CELT_SET_VBR_RATE_REQUEST, _celt_check_int(x)
- #define CELT_RESET_STATE_REQUEST 8
- #define CELT_GET_FRAME_SIZE 1000
- #define CELT_GET_LOOKAHEAD 1001
- #define CELT_GET_SAMPLE_RATE 1003
- #define CELT_GET_BITSTREAM_VERSION 2000

Typedefs

- typedef struct CELTEncoder CELTEncoder

  Encoder state.

- typedef struct CELTDecoder CELTDecoder
- typedef struct CELTMode CELTMode

Functions

- EXPORT CELTMode ∗ celt_mode_create (celt_int32 Fs, int frame_size, int ∗error)
- EXPORT void celt_mode_destroy (CELTMode ∗mode)
- EXPORT int celt_mode_info (const CELTMode ∗mode, int request, celt_int32 ∗value)
- EXPORT CELTEncoder ∗ celt_encoder_create (const CELTMode ∗mode, int channels, int ∗error)
- EXPORT void celt_encoder_destroy (CELTEncoder ∗st)
- EXPORT int celt_encode_float (CELTEncoder ∗st, const float ∗pcm, float ∗optional_synthesis, unsigned char ∗compressed, int nbCompressedBytes)
- EXPORT int celt_encode (CELTEncoder ∗st, const celt_int16 ∗pcm, celt_int16 ∗optional_synthesis, unsigned char ∗compressed, int nbCompressedBytes)
- EXPORT int celt_encoder_ctl (CELTEncoder ∗st, int request,...)
- EXPORT CELTDecoder ∗ celt_decoder_create (const CELTMode ∗mode, int channels, int ∗error)
- EXPORT void celt_decoder_destroy (CELTDecoder ∗st)
- EXPORT int celt_decode_float (CELTDecoder ∗st, const unsigned char ∗data, int len, float ∗pcm)
- EXPORT int celt_decode (CELTDecoder ∗st, const unsigned char ∗data, int len, celt_int16 ∗pcm)
- EXPORT int celt_decoder_ctl (CELTDecoder ∗st, int request,...)
- EXPORT const char ∗ celt_strerror (int error)

6.1.1 Detailed Description

Contains all the functions for encoding and decoding audio.

Definition in file celt.h.

6.1.2 Define Documentation

6.1.2.1 #define CELT_ALLOC_FAIL -7

Memory allocation has failed

Definition at line 75 of file celt.h.

6.1.2.2 #define CELT_BAD_ARG -1

An (or more) invalid argument (e.g. out of range)

Definition at line 63 of file celt.h.

6.1.2.3 #define CELT_CORRUPTED_DATA -4

The data passed (e.g. compressed data to decoder) is corrupted

Definition at line 69 of file celt.h.

6.1.2.4 #define CELT_GET_BITSTREAM_VERSION 2000

GET the bit-stream version for compatibility check

Definition at line 106 of file celt.h.

6.1.2.5 #define CELT_GET_FRAME_SIZE 1000

GET the frame size used in the current mode

Definition at line 99 of file celt.h.

6.1.2.6 #define CELT_GET_LOOKAHEAD 1001

GET the lookahead used in the current mode

Definition at line 101 of file celt.h.

6.1.2.7 #define CELT_GET_MODE(x) CELT_GET_MODE_REQUEST, _celt_check_mode_ptr_ptr(x)

Get the CELTMode used by an encoder or decoder

Definition at line 80 of file celt.h.

6.1.2.8 #define CELT_GET_SAMPLE_RATE 1003

GET the sample rate used in the current mode

Definition at line 103 of file celt.h.

6.1.2.9 #define CELT_INTERNAL_ERROR -3

An internal error was detected

Definition at line 67 of file celt.h.

6.1.2.10 #define CELT_INVALID_MODE -2

The mode struct passed is invalid

Definition at line 65 of file celt.h.

6.1.2.11 #define CELT_INVALID_STATE -6

An encoder or decoder structure is invalid or already freed

Definition at line 73 of file celt.h.

6.1.2.12 #define CELT_OK 0

No error

Definition at line 61 of file celt.h.

6.1.2.13 #define CELT_RESET_STATE_REQUEST 8

Reset the encoder/decoder memories to zero

Definition at line 95 of file celt.h.

6.1.2.14 #define CELT_SET_COMPLEXITY(x) CELT_SET_COMPLEXITY_REQUEST, _celt_check_int(x)

Controls the complexity from 0-10 (int)

Definition at line 83 of file celt.h.

6.1.2.15 #define CELT_SET_PREDICTION(x) CELT_SET_PREDICTION_REQUEST, _celt_check_int(x)

Controls the use of interframe prediction. 0=Independent frames 1=Short term interframe prediction allowed 2=Long term prediction allowed

Definition at line 90 of file celt.h.

6.1.2.16 #define CELT_SET_VBR_RATE(x) CELT_SET_VBR_RATE_REQUEST, _celt_check_int(x)

Set the target VBR rate in bits per second(int); 0=CBR (default)

Definition at line 93 of file celt.h.

6.1.2.17 #define CELT_UNIMPLEMENTED -5

Invalid/unsupported request number

Definition at line 71 of file celt.h.

6.1.3 Typedef Documentation

6.1.3.1 typedef struct CELTDecoder CELTDecoder

State of the decoder. One decoder state is needed for each stream. It is initialised once at the beginning of the stream. Do ∗not∗ re-initialise the state for every frame

Definition at line 119 of file celt.h.

6.1.3.2 typedef struct CELTEncoder CELTEncoder

Encoder state. Contains the state of an encoder. One encoder state is needed for each stream. It is initialised once at the beginning of the stream. Do ∗not∗ re-initialise the state for every frame.

Definition at line 114 of file celt.h.

6.1.3.3 typedef struct CELTMode CELTMode

The mode contains all the information necessary to create an encoder. Both the encoder and decoder need to be initialised with exactly the same mode, otherwise the quality will be very bad

Definition at line 125 of file celt.h.

6.2 libcelt/celt_types.h File Reference

CELT types.

6.2.1 Detailed Description

CELT types.

Definition in file celt_types.h.

Index