Reference Manual

Generated by Doxygen 1.7.4

Wed Aug 31 2011 13:54:26

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 _MAGE_Environment Struct Reference

Public Attributes

- MAGE_Engine * engine
- void * data

3.1.1 Member Data Documentation

3.1.1.1 void* _MAGE_Environment::data

An MAGE_Environment variable.

Void pointer to to any non-engine specific data used.

3.1.1.2 MAGE_Engine* _MAGE_Environment::engine

An MAGE_Environment variable.

MAGE_Engine pointer to the engine used.

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

- MAGE/MAGE.h

3.2 _pHTS Struct Reference

Public Attributes

- FILE * p_lf0fp

- FILE * p_mcpfp
- FILE * p_durfp
- FILE * p_tracefp
- FILE * p_rawfp
- FILE * p_wavfp
- FILE * p_labels
- int num_interp
- double * rate_interp
- char ** fn_ms_lf0
- char ** fn_ms_mcp
- char ** fn_ms_dur
- int num_ms_lf0
- int num_ms_mcp
- int num_ms_dur
- char ** fn_ts_lf0
- char ** fn_ts_mcp
- char ** fn_ts_dur
- int num_ts_lf0
- int num_ts_mcp
- int num_ts_dur
- char ** fn_ws_lf0
- char ** fn_ws_mcp
- int num_ws_lf0
- int num_ws_mcp
- char ** fn_ms_gvl
- char ** fn_ms_gvm
- int num_ms_gvl
- int num_ms_gvm
- int sampling_rate
- int fperiod
- double alpha
- double stage
- double beta
- double uv_threshold
- double gv_weight_lf0
- double gv_weight_mcp
- double f0_std
- double f0_mean
- HTS_Boolean phoneme_alignment
- double speech_speed
- HTS_Boolean use_log_gain
- char * labfn
- int delay
- double pitch
- int pitch_mode
- double duration

- int duration_mode
- double volume
- HTS_Boolean reset
- HTS_Vocoder vocoder
- int * total_frame_array
- int * length_array
- RingBuffer * labelRingBuffer
- RingBuffer * audioRingBuffer

3.2.1 Member Data Documentation

3.2.1.1 double _pHTS::alpha

An pHTS environment variable.

Double for the all-pass constant.

Attention

Default value is 0.42.

3.2.1.2 RingBuffer* _pHTS::audioRingBuffer

An pHTS environment variable.

RingBuffer for storing the speech samples before the audio callback().

3.2.1.3 double _pHTS::beta

An pHTS environment variable.

Double for the postfiltering coefficient.

Attention

Default value is 0.0. If you set beta large value, formant structure will be emphasized strongly.

3.2.1.4 int _pHTS::delay

An pHTS environment variable.

Integer for the number of future phonetic labels that will be used for the co-articulation. If delay < 0 the pHTS behaves in the same way as the original HTS, waiting for all the phonemes to arrive before synthesising the speech samples.

3.2.1.5 double _pHTS::duration

An pHTS environment variable.

Double for the duration value to be used in the duration trajectory of the engine.

3.2.1.6 int _pHTS::duration_mode

An pHTS environment variable.

Integer to determine the action applied on the duration trajectory, if duration_mode=0, the existing trajectory is overwritten with the value of duration, if duration_mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if duration_mode=2, the existing trajectory is scaled.

3.2.1.7 double _pHTS::f0_mean

An pHTS environment variable.

Double for the standard deviation for the spectrum.

3.2.1.8 double _pHTS::f0_std

An pHTS environment variable.

Double for the standard deviation for the log f0.

3.2.1.9 char** _pHTS::fn_ms_dur

An pHTS environment variable.

Char pointer to the file name where the duration model is stored.

Attention

Mandatory as a command line argument.

3.2.1.10 char** _pHTS::fn_ms_gvl

An pHTS environment variable.

Char pointer to the file name where the global variance for log f0 is stored.

Attention

Mandatory as a command line argument.

3.2.1.11 char** _pHTS::fn_ms_gvm

An pHTS environment variable.

Char pointer to the file name where the global variance for spectrum is stored.

Attention

Mandatory as a command line argument.

3.2.1.12 char** _pHTS::fn_ms_lf0

An pHTS environment variable.

Char pointer to the file name where the log f0 model is stored.

Attention

Mandatory as a command line argument.

3.2.1.13 char** _pHTS::fn_ms_mcp

An pHTS environment variable.

Char pointer to the file name where the spectrum model is stored.

Attention

Mandatory as a command line argument.

3.2.1.14 char** _pHTS::fn_ts_dur

An pHTS environment variable.

Char pointer to the file name where the duration tree is stored.

Attention

Mandatory as a command line argument.

3.2.1.15 char** _pHTS::fn_ts_lf0

An pHTS environment variable.

Char pointer to the file name where the log f0 tree is stored.

Attention

Mandatory as a command line argument.

3.2.1.16 char** _pHTS::fn_ts_mcp

An pHTS environment variable.

Char pointer to the file name where the spectrum tree is stored.

Attention

Mandatory as a command line argument.

3.2.1.17 char** _pHTS::fn_ws_lf0

An pHTS environment variable.

Char pointer to the file name where the log f0 windows are stored.

Attention

Mandatory as a command line argument.

3.2.1.18 char** _pHTS::fn_ws_mcp

An pHTS environment variable.

Char pointer to the file name where the spectrum windows are stored.

Attention

Mandatory as a command line argument.

3.2.1.19 int _pHTS::fperiod

An pHTS environment variable.

Integer for the frame period.

Attention

Default value is 80.

3.2.1.20 double _pHTS::gv_weight_lf0

An pHTS environment variable.

Double for the global variance weight for the log f0.

3.2.1.21 double _pHTS::gv_weight_mcp

An pHTS environment variable.

Double for the global variance weight for the spectrum.

3.2.1.22 RingBuffer* _pHTS::labelRingBuffer

An pHTS environment variable.

RingBuffer for storing the incomming labels to be processed.

3.2.1.23 char* _pHTS::labfn

An pHTS environment variable.

Char pointer to the file name where the precomputed labels are stored, in order to be loaded.

Attention

Mandatory as a command line argument.

3.2.1.24 int* _pHTS::length_array

An pHTS environment variable.

Integer array for the length of all the streams.

3.2.1.25 int _pHTS::num_interp

An pHTS environment variable.

Integer for the number of the speakers for interpolation. Not mandatory as a command line argument.

3.2.1.26 int _pHTS::num_ms_dur

An pHTS environment variable.

Integer for the number of the duration models for interpolation.

3.2.1.27 int _pHTS::num_ms_gvl

An pHTS environment variable.

Integer for the number of the global variance for log f0.

3.2.1.28 int _pHTS::num_ms_gvm

An pHTS environment variable.

Integer for the number of the global variance for spectrum.

3.2.1.29 int _pHTS::num_ms_lf0

An pHTS environment variable.

Integer for the number of the log f0 models for interpolation.

3.2.1.30 int _pHTS::num_ms_mcp

An pHTS environment variable.

Integer for the number of the spectrum models for interpolation.

3.2.1.31 int _pHTS::num_ts_dur

An pHTS environment variable.

Integer for the number of the duration trees for interpolation.

3.2.1.32 int _pHTS::num_ts_lf0

An pHTS environment variable.

Integer for the number of the log f0 trees for interpolation.

3.2.1.33 int _pHTS::num_ts_mcp

An pHTS environment variable.

Integer for the number of the spectrum trees for interpolation.

3.2.1.34 int _pHTS::num_ws_lf0

An pHTS environment variable.

Integer for the number of the log f0 windows.

3.2.1.35 int _pHTS::num_ws_mcp

An pHTS environment variable.

Integer for the number of the spectrum windows.

3.2.1.36 FILE* _pHTS::p_durfp

An pHTS environment variable.

File pointer to the file where the produced labels with time will be saved.

3.2.1.37 FILE* _pHTS::p_labels

An pHTS environment variable.

File pointer to the file where the precomputed labels are stored, in order to be loaded.

3.2.1.38 FILE* _pHTS::p_lf0fp

An pHTS environment variable.

File pointer to the file where the produced log f0 will be saved.

3.2.1.39 FILE* _pHTS::p_mcpfp

An pHTS environment variable.

File pointer to the file where the produced spectrum will be saved.

3.2.1.40 FILE* _pHTS::p_rawfp

An pHTS environment variable.

File pointer to the file where the produced speech samples will be saved.

3.2.1.41 FILE* _pHTS::p_tracefp

An pHTS environment variable.

File pointer to the file where the general engine information will be saved.

3.2.1.42 FILE* _pHTS::p_wavfp

An pHTS environment variable.

File pointer to the file where the produced wave file from the speech samples will be saved.

3.2.1.43 HTS_Boolean _pHTS::phoneme_alignment

An pHTS environment variable.

Integer for the alignment of the phonemes (0 for false or 1 for true).

3.2.1.44 double _pHTS::pitch

An pHTS environment variable.

Double for the new pitch value to be used in the pitch trajectory of the engine.

3.2.1.45 int _pHTS::pitch_mode

An pHTS environment variable.

Integer to determine the action applied on the pitch trajectory, if pitch_mode=0, the existing trajectory is overwritten with the value of pitch, if pitch_mode=1, the existing trajectory is shifted up if pitch$>$0 or shifted down if pitch$<$0.

3.2.1.46 double* _pHTS::rate_interp

An pHTS environment variable.

Integer array for the speakers interpolation. Not mandatory as a command line argument.

3.2.1.47 HTS_Boolean _pHTS::reset

An pHTS environment variable.

Integer for the number of the phonetic labels that will be kept im memory before refreshing the engine. if reset == FALSE then the delayed labels will not be estimated/synthesised, if reset == TRUE, then for each # labels + the delayed labels will be estimated/synthesised.

3.2.1.48 int _pHTS::sampling_rate

An pHTS environment variable.

Integer for the sampling rate.

Attention

Default value is 16000.

3.2.1.49 double _pHTS::speech_speed

An pHTS environment variable.

Double for the speech of the produced speech.

3.2.1.50 double _pHTS::stage

An pHTS environment variable.

Double for the stage.

Attention

Default value is 0.

3.2.1.51 int* _pHTS::total_frame_array

An pHTS environment variable.

Integer array for the number of frames for all the labels.

3.2.1.52 HTS_Boolean _pHTS::use_log_gain

An pHTS environment variable.

Integer for use the log gain (0 for false or 1 for true).

3.2.1.53 double _pHTS::uv_threshold

An pHTS environment variable.

Double for the voiced/unvoiced threshold.

3.2.1.54 HTS_Vocoder _pHTS::vocoder

An pHTS environment variable.

HTS_Vocoder for the synthesis of the speech samples.

3.2.1.55 double _pHTS::volume

An pHTS environment variable.

Double for the volume value to be used in the generation of the speech samples.

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

- pHTS_lib/pHTS.h

3.3 MAGE_Environment Struct Reference

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

- MAGE/MAGE.h

3.4 pHTS Struct Reference

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

- pHTS_lib/pHTS.h

Chapter 4

File Documentation

4.1 MAGE/MAGE.c File Reference

Contains the implementation of all the functions of the MAGE environment. It is the threading interface of pHTS.

```
#include "MAGE.h"
```

Functions

- MAGE_Environment * MAGE_init (int argc, char **argv)
- void MAGE_pushLabel (MAGE_Environment *environment, char *labelAsStr)
- int MAGE_parseLabel (MAGE_Environment *environment)
- void MAGE_setSpeed (MAGE_Environment *environment, double speechSpeed)
- double MAGE_getSpeed (MAGE_Environment *environment)
- void MAGE_setPitch (MAGE_Environment *environment, double pitch, int mode)
- double MAGE_getPitch (MAGE_Environment *environment)
- void MAGE_setVolume (MAGE_Environment *environment, double volume)
- double MAGE_getVolume (MAGE_Environment *environment)
- void MAGE_setAlpha (MAGE_Environment *environment, double alpha)
- double MAGE_getAlpha (MAGE_Environment *environment)
- void MAGE_setDuration (MAGE_Environment *environment, double duration, int mode)
- double MAGE_getDuration (MAGE_Environment *environment)
- void MAGE_update (MAGE_Environment *environment)
- void MAGE_updatePDFs (MAGE_Environment *environment)
- void MAGE_updateFilter (MAGE_Environment *environment)
- void MAGE_updateSamples (MAGE_Environment *environment)
- int MAGE_getNumberOfLabels (MAGE_Environment *environment)
- int MAGE_getNumberOfSamples (MAGE_Environment *environment)
- MAGE_Label * MAGE_getLabel (MAGE_Environment *environment)
- MAGE_PDFs * MAGE_getPDFs (MAGE_Environment *environment)

- void MAGE_setPDFs (MAGE_Environment *environment, MAGE_PDFs *pdfs)
- MAGE_Filter * MAGE_getFilter (MAGE_Environment *environment)
- void MAGE_setFilter (MAGE_Environment *environment, MAGE_Filter *filter)
- void MAGE_getSamples (MAGE_Environment *environment, short *buffer, int nOfSamples)
- void MAGE_popSamples (MAGE_Environment *environment, float *buffer, int nOfSamples)
- void MAGE_finalize (MAGE_Environment *environment)
- void MAGE_refresh (MAGE_Environment *environment)
- void MAGE_free (MAGE_Environment *environment)
- void * MAGE_getLabelFromFile (MAGE_Environment *environment)

4.1.1 Detailed Description

Contains the implementation of all the functions of the MAGE environment. It is the threading interface of pHTS.

4.1.2 Function Documentation

4.1.2.1 void MAGE_finalize (MAGE_Environment * *environment*)

A MAGE_Environment method to to flush the delayed labels.

Parameters

environment	a MAGE_Environment pointer to be flushed.

4.1.2.2 void MAGE_free (MAGE_Environment * *environment*)

A MAGE_Environment method to free all the alocated memory occupied by the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be freed.

4.1.2.3 double MAGE_getAlpha (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current alpha, all-pass constant value from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current alpha, all-pass constant value from the MAGE_Environment.

4.1.2.4 double MAGE_getDuration (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current duration from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current duration from the MAGE_Environment.

4.1.2.5 MAGE_Filter* MAGE_getFilter (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current generated speech parameters set of the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

a void pointer to the current generated speech parameters set of the engine of the MAGE_Environment.

4.1.2.6 MAGE_Label* MAGE_getLabel (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current label processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

a void pointer to the current label processed by the engine of the MAGE_Environment.

4.1.2.7 void* MAGE_getLabelFromFile (MAGE_Environment * *environment*)

A MAGE_Environment method to get one label string from a label file.

Parameters

environment	a MAGE_Environment pointer to be freed.

Returns

a pointer to the new label string retrieved from the label file.

4.1.2.8 int MAGE_getNumberOfLabels (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current number of labels queued in the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

an integer with the current number of labels queued in the engine of the MAGE_Environment.

4.1.2.9 int MAGE_getNumberOfSamples (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current number of generated samples queued in the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

an integer with the current number of generated samples queued in the engine of the MAGE_Environment.

4.1.2.10 MAGE_PDFs* MAGE_getPDFs (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current PDFs set processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

a void pointer to the current PDFs set processed by the engine of the MAGE_Environment.

4.1.2.11 double MAGE_getPitch (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current pitch from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current pitch from the MAGE_Environment.

4.1.2.12 void MAGE_getSamples (MAGE_Environment * *environment*, short * *buffer*, int *nOfSamples*)

A MAGE_Environment method to acces the generated speech samples.

Parameters

environment	a MAGE_Environment pointer to be accessed.
buffer	a short array to store the generated speech samples.
nOfSamples	an integer for the number of dumples to be accessed.

4.1.2.13 double MAGE_getSpeed (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current speech speed from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current speech speed from the MAGE_Environment.

4.1.2.14 double MAGE_getVolume (MAGE_Environment * *environment*)

A MAGE_Environment method to get the volume value from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current volume value from the MAGE_Environment.

4.1.2.15 MAGE_Environment* MAGE_init (int *argc*, char ** *argv*)

A MAGE_Environment method to initialize the MAGE_Environment and returns its pointer.

Parameters

argc	an integer for the number of arguments.
argv	a character pointer to all arguments.

Returns

the initialized MAGE_Environment pointer.

4.1.2.16 int MAGE_parseLabel (MAGE_Environment * *environment*)

A MAGE_Environment method to parse the the last pushed label in the list of labels of the MAGE_Environment and determine state duration.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.1.2.17 void MAGE_popSamples (MAGE_Environment * *environment*, float * *buffer*, int *nOfSamples*)

A MAGE_Environment method to access and pop the generated speech samples from MAGE ring buffer.

Parameters

environment	a MAGE_Environment pointer to be accessed.
buffer	float pointer to the buffer to be filled with the generated samples.
nOfSamples	an integer for the number of samples to be poped out of the ring buffer.

Attention

Must be called from audio callback

4.1.2.18 void MAGE_pushLabel (MAGE_Environment * *environment*, char * *labelAsStr*)

A MAGE_Environment method to push a new string label recevied from the program (from a file or on the fly) onto the label queue of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.
labelAsStr	character pointer to a phonetic label.

4.1.2.19 void MAGE_refresh (MAGE_Environment * *environment*)

A MAGE_Environment method to flush the internal memory of the engine occupied by the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be flushed.

4.1.2.20 void MAGE_setAlpha (MAGE_Environment * *environment,* double *alpha*)

A MAGE_Environment method to set the current alpha, all-pass constant value in the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.
alpha	a double for the new alpha value.

4.1.2.21 void MAGE_setDuration (MAGE_Environment * *environment,* double *duration,* int *mode*)

A MAGE_Environment method to set the duration in the MAGE_Environment to a new value.

Parameters

environment	a MAGE_Environment pointer to be updated.
duration	a double for the new duration value.
mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

4.1.2.22 void MAGE_setFilter (MAGE_Environment * *environment,* MAGE_Filter * *filter*)

A MAGE_Environment method to override the current set of generated speech parameters processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.
filter	a MAGE_Filter pointer to override the current set of generated speech parameters.

4.1.2.23 void MAGE_setPDFs (MAGE_Environment * *environment*, MAGE_PDFs * *pdfs*)

A MAGE_Environment method to override the current PDFs set processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.
pdfs	a MAGE_PDFs pointer to override the current PDFs set.

4.1.2.24 void MAGE_setPitch (MAGE_Environment * *environment*, double *pitch*, int *mode*)

A MAGE_Environment method to set the pitch in the MAGE_Environment to a new value.

Parameters

environment	a MAGE_Environment pointer to be updated.
pitch	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of pitch, if mode=1, the existing trajectory is shifted up if pitch>0 or shifted down if pitch<0.

4.1.2.25 void MAGE_setSpeed (MAGE_Environment * *environment*, double *speechSpeed*)

A MAGE_Environment method to set the speech speed rate in the MAGE_Environment to a new value.

Parameters

environment	a MAGE_Environment pointer to be updated.
speech-Speed	a double for the new speech speed rate.

4.1.2.26 void MAGE_setVolume (MAGE_Environment * *environment*, double *volume*)

A MAGE_Environment method to set the volume value in the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.
volume	a double for the new volume value.

4.1.2.27 void MAGE_update (MAGE_Environment * *environment*)

A MAGE_Environment method to generate the speech samples. This function is a high-level call for converting first queued label and current prosody into sound.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.1.2.28 void MAGE_updateFilter (MAGE_Environment * *environment*)

A MAGE_Environment method to convert current PDFs into speech parameters.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.1.2.29 void MAGE_updatePDFs (MAGE_Environment * *environment*)

A MAGE_Environment method to convert first queued label into PDFs.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.1.2.30 void MAGE_updateSamples (MAGE_Environment * *environment*)

A MAGE_Environment method to generate speech samples from the speech parameters.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.2 MAGE/MAGE.h File Reference

Contains the decleration of all the functions of the threading MAGE_Environment.

```
#include "pHTS.h"
```

Classes

- struct _MAGE_Environment

Defines

- #define MAGE_OVERWRITE 0

  A macro that defines overwriting as the opperation to be applied.
- #define MAGE_SHIFT 1

  A macro that defines shifting as the opperation to be applied.
- #define MAGE_SCALE 2

  A macro that defines scaling as the opperation to be applied.

Typedefs

- typedef void MAGE_Engine

  A generic type for identifiying the engine used in the MAGE_Environment.
- typedef void MAGE_Label

  A type definition for the Labels used in MAGE_Environment.
- typedef void MAGE_PDFs

  A type definition for the PDFs used in MAGE_Environment.
- typedef void MAGE_Filter

  A type definition for the generated speech parameters used in MAGE_Environment.
- typedef struct _MAGE_Environment MAGE_Environment

  A type definition for the MAGE_Environment, that is the main data type that is shared through all MAGE functions.

Functions

- MAGE_Environment * MAGE_init (int argc, char **argv)
- void MAGE_pushLabel (MAGE_Environment *environment, char *labelAsStr)
- int MAGE_parseLabel (MAGE_Environment *environment)
- void MAGE_setSpeed (MAGE_Environment *environment, double speechSpeed)
- double MAGE_getSpeed (MAGE_Environment *environment)
- void MAGE_setPitch (MAGE_Environment *environment, double pitch, int mode)
- double MAGE_getPitch (MAGE_Environment *environment)
- void MAGE_setAlpha (MAGE_Environment *environment, double alpha)
- double MAGE_getAlpha (MAGE_Environment *environment)
- void MAGE_setVolume (MAGE_Environment *environment, double volume)
- double MAGE_getVolume (MAGE_Environment *environment)
- void MAGE_setDuration (MAGE_Environment *environment, double duration, int mode)
- double MAGE_getDuration (MAGE_Environment *environment)
- void MAGE_update (MAGE_Environment *environment)
- void MAGE_updatePDFs (MAGE_Environment *environment)
- void MAGE_updateFilter (MAGE_Environment *environment)
- void MAGE_updateSamples (MAGE_Environment *environment)
- int MAGE_getNumberOfLabels (MAGE_Environment *environment)
- int MAGE_getNumberOfSamples (MAGE_Environment *environment)

- MAGE_Label * MAGE_getLabel (MAGE_Environment *environment)
- MAGE_PDFs * MAGE_getPDFs (MAGE_Environment *environment)
- void MAGE_setPDFs (MAGE_Environment *environment, MAGE_PDFs *pdfs)
- MAGE_Filter * MAGE_getFilter (MAGE_Environment *environment)
- void MAGE_setFilter (MAGE_Environment *environment, MAGE_Filter *filter)
- void MAGE_getSamples (MAGE_Environment *environment, short *buffer, int nOfSamples)
- void MAGE_popSamples (MAGE_Environment *environment, float *buffer, int nOfSamples)
- void MAGE_finalize (MAGE_Environment *environment)
- void MAGE_refresh (MAGE_Environment *environment)
- void MAGE_free (MAGE_Environment *environment)
- void * MAGE_getLabelFromFile (MAGE_Environment *environment)

4.2.1 Detailed Description

Contains the decleration of all the functions of the threading MAGE_Environment.

4.2.2 Typedef Documentation

4.2.2.1 typedef struct _MAGE_Environment MAGE_Environment

A type definition for the MAGE_Environment, that is the main data type that is shared through all MAGE functions.

The MAGE_Environment struct contains all the passed arguments for the MAGE_-Engine and the variables needed for the threading environment.

4.2.3 Function Documentation

4.2.3.1 void MAGE_finalize (MAGE_Environment * *environment*)

A MAGE_Environment method to to flush the delayed labels.

Parameters

environment	a MAGE_Environment pointer to be flushed.

4.2.3.2 void MAGE_free (MAGE_Environment * *environment*)

A MAGE_Environment method to free all the alocated memory occupied by the MAGE_-Environment.

Parameters

environment	a MAGE_Environment pointer to be freed.

4.2.3.3 double MAGE_getAlpha (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current alpha, all-pass constant value from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current alpha, all-pass constant value from the MAGE_Environment.

4.2.3.4 double MAGE_getDuration (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current duration from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current duration from the MAGE_Environment.

4.2.3.5 MAGE_Filter* MAGE_getFilter (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current generated speech parameters set of the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

a void pointer to the current generated speech parameters set of the engine of the MAGE_Environment.

4.2.3.6 MAGE_Label* MAGE_getLabel (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current label processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

a void pointer to the current label processed by the engine of the MAGE_Environment.

4.2.3.7 void* MAGE_getLabelFromFile (MAGE_Environment * *environment*)

A MAGE_Environment method to get one label string from a label file.

Parameters

environment	a MAGE_Environment pointer to be freed.

Returns

a pointer to the new label string retrieved from the label file.

4.2.3.8 int MAGE_getNumberOfLabels (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current number of labels queued in the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

an integer with the current number of labels queued in the engine of the MAGE_-Environment.

4.2.3.9 int MAGE_getNumberOfSamples (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current number of generated samples queued in the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

an integer with the current number of generated samples queued in the engine of the MAGE_Environment.

4.2.3.10 MAGE_PDFs* MAGE_getPDFs (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current PDFs set processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.

Returns

a void pointer to the current PDFs set processed by the engine of the MAGE_Environment.

4.2.3.11 double MAGE_getPitch (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current pitch from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current pitch from the MAGE_Environment.

4.2.3.12 void MAGE_getSamples (MAGE_Environment * *environment*, short * *buffer*, int *nOfSamples*)

A MAGE_Environment method to acces the generated speech samples.

Parameters

environment	a MAGE_Environment pointer to be accessed.
buffer	a short array to store the generated speech samples.
nOfSamples	an integer for the number of dumples to be accessed.

4.2.3.13 double MAGE_getSpeed (MAGE_Environment * *environment*)

A MAGE_Environment method to get the current speech speed from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current speech speed from the MAGE_Environment.

4.2.3.14 double MAGE_getVolume (MAGE_Environment * *environment*)

A MAGE_Environment method to get the volume value from the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.

Returns

a double with the current volume value from the MAGE_Environment.

4.2.3.15 MAGE_Environment* MAGE_init (int *argc*, char ** *argv*)

A MAGE_Environment method to initialize the MAGE_Environment and returns its pointer.

Parameters

argc	an integer for the number of arguments.
argv	a character pointer to all arguments.

Returns

the initialized MAGE_Environment pointer.

4.2.3.16 int MAGE_parseLabel (MAGE_Environment * *environment*)

A MAGE_Environment method to parse the the last pushed label in the list of labels of the MAGE_Environment and determine state duration.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.2.3.17 void MAGE_popSamples (MAGE_Environment * *environment*, float * *buffer*, int *nOfSamples*)

A MAGE_Environment method to access and pop the generated speech samples from MAGE ring buffer.

Parameters

environment	a MAGE_Environment pointer to be accessed.
buffer	float pointer to the buffer to be filled with the generated samples.
nOfSamples	an integer for the number of samples to be poped out of the ring buffer.

Attention

Must be called from audio callback

4.2.3.18 void MAGE_pushLabel (MAGE_Environment * *environment,* char * *labelAsStr*)

A MAGE_Environment method to push a new string label recevied from the program (from a file or on the fly) onto the label queue of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.
labelAsStr	character pointer to a phonetic label.

4.2.3.19 void MAGE_refresh (MAGE_Environment * *environment*)

A MAGE_Environment method to flush the internal memory of the engine occupied by the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be flushed.

4.2.3.20 void MAGE_setAlpha (MAGE_Environment * *environment,* double *alpha*)

A MAGE_Environment method to set the current alpha, all-pass constant value in the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.
alpha	a double for the new alpha value.

4.2.3.21 void MAGE_setDuration (MAGE_Environment * *environment,* double *duration,* int *mode*)

A MAGE_Environment method to set the duration in the MAGE_Environment to a new value.

Parameters

environment	a MAGE_Environment pointer to be updated.
duration	a double for the new duration value.
mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

4.2.3.22 void MAGE_setFilter (MAGE_Environment * *environment*, MAGE_Filter * *filter*)

A MAGE_Environment method to override the current set of generated speech parameters processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.
filter	a MAGE_Filter pointer to override the current set of generated speech parameters.

4.2.3.23 void MAGE_setPDFs (MAGE_Environment * *environment*, MAGE_PDFs * *pdfs*)

A MAGE_Environment method to override the current PDFs set processed by the engine of the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be accessed.
pdfs	a MAGE_PDFs pointer to override the current PDFs set.

4.2.3.24 void MAGE_setPitch (MAGE_Environment * *environment*, double *pitch*, int *mode*)

A MAGE_Environment method to set the pitch in the MAGE_Environment to a new value.

Parameters

environment	a MAGE_Environment pointer to be updated.
pitch	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of pitch, if mode=1, the existing trajectory is shifted up if pitch>0 or shifted down if pitch<0.

4.2.3.25 void MAGE_setSpeed (MAGE_Environment * *environment*, double *speechSpeed*)

A MAGE_Environment method to set the speech speed rate in the MAGE_Environment to a new value.

Parameters

environment	a MAGE_Environment pointer to be updated.
speechSpeed	a double for the new speech speed rate.

4.2.3.26 void MAGE_setVolume (MAGE_Environment * *environment*, double *volume*)

A MAGE_Environment method to set the volume value in the MAGE_Environment.

Parameters

environment	a MAGE_Environment pointer to be updated.
volume	a double for the new volume value.

4.2.3.27 void MAGE_update (MAGE_Environment * *environment*)

A MAGE_Environment method to generate the speech samples. This function is a high-level call for converting first queued label and current prosody into sound.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.2.3.28 void MAGE_updateFilter (MAGE_Environment * *environment*)

A MAGE_Environment method to convert current PDFs into speech parameters.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.2.3.29 void MAGE_updatePDFs (MAGE_Environment * *environment*)

A MAGE_Environment method to convert first queued label into PDFs.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.2.3.30 void MAGE_updateSamples (MAGE_Environment * *environment*)

A MAGE_Environment method to generate speech samples from the speech parameters.

Parameters

environment	a MAGE_Environment pointer to be updated.

4.3 pHTS_lib/p_HTS_engine.c File Reference

Contains the implementation of all the streaming functions of the HTS_Engine.

```
#include "HTS_hidden.h"
```

Functions

- void p_HTS_Engine_load_label_from_str (HTS_Engine ∗engine, char ∗label_str)
- void p_HTS_Engine_load_label_from_fp (HTS_Engine ∗engine, FILE ∗fp)
- void p_HTS_Engine_create_sstream (HTS_Engine ∗engine, int ∗∗total_frame_array, double duration, int mode)
- void p_HTS_Engine_create_pstream (HTS_Engine ∗engine, int ∗total_frame_array, int ∗∗length_array, int past)
- void p_HTS_Engine_create_gstream_param (HTS_Engine ∗engine, int delay, int ∗total_frame_array, int ∗length_array, HTS_Boolean reset)
- void p_HTS_Engine_create_gstream_speech_samples (HTS_Engine ∗engine, int delay, HTS_Vocoder ∗v, int ∗total_frame_array)
- void p_HTS_Engine_set_labels (HTS_Engine ∗engine, HTS_Label ∗label)
- void p_HTS_Engine_set_pstream (HTS_Engine ∗engine, HTS_PStreamSet ∗pss)
- void p_HTS_Engine_set_gstream (HTS_Engine ∗engine, HTS_GStreamSet ∗gss)
- void p_HTS_Engine_set_gstream_pitch (HTS_Engine ∗engine, int delay, int ∗total_frame_array, double pitch, int mode)

4.3.1 Detailed Description

Contains the implementation of all the streaming functions of the HTS_Engine. The functions implemented here, are the streaming version of the original HTS_Engine functions.

4.3.2 Function Documentation

4.3.2.1 void p_HTS_Engine_create_gstream_param (HTS_Engine ∗ *engine*, int *delay*, int ∗ *total_frame_array*, int ∗ *length_array*, HTS_Boolean *reset*)

A streaming engine method to generate speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
delay	an integer for the number of labels used as delay, number of used future labels.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.
reset	a boolean integer for the number of labels to be synthesised before refreshing the engine. If reset=0 then the delayed frames are not estimated/synthesised, if reset=1 then the delayed framd are estimated/synthesised.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.
To synthesize speech, you must set stream[0]=spectrum models and spectrum[1]=lf0 models.

4.3.2.2 void p_HTS_Engine_create_gstream_speech_samples (HTS_Engine * *engine*, int *delay*, HTS_Vocoder * *v*, int * *total_frame_array*)

A streaming engine method to generate speech samples.

Parameters

engine	a HTS_Engine pointer to be updated.
delay	an integer for the number of labels used as delay, number of used future labels.
v	a HTS_Vocoder pointer to the structure for setting of vocoder.
total_frame_array	a pointer to the array with the total number of frames.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.
To synthesize speech, you must set stream[0]=spectrum models and spectrum[1]=lf0 models.

4.3.2.3 void p_HTS_Engine_create_pstream (HTS_Engine * *engine*, int * *total_frame_array*, int ** *length_array*, int *past*)

A streaming engine method to generate parameter using GV weight.

Parameters

engine	a HTS_Engine pointer to be updated.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.
past	an integer for the number of past labels to be used.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.3.2.4 void p_HTS_Engine_create_sstream (HTS_Engine * *engine*, int ** *total_frame_array*, double *duration*, int *mode*)

A streaming engine method to parse label and determine state duration.

Parameters

engine	a HTS_Engine pointer to be updated.
total_frame_array	a pointer to the array with the total number of frames.
duration	a double for the duration of the phoneme in the label
mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.3.2.5 void p_HTS_Engine_load_label_from_fp (HTS_Engine * *engine*, FILE * *fp*)

A streaming engine method to load a phonetic label from file pointer.

Parameters

engine	a HTS_Engine pointer to be updated.
fp	a file pointer to the file containing phonetic labels.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.3.2.6 void p_HTS_Engine_load_label_from_str (HTS_Engine * *engine*, char * *label_str*)

A streaming engine method to load a phonetic label from a string.

Parameters

engine	a HTS_Engine pointer to be updated.
label_str	character pointer to a phonetic label.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.3.2.7 void p_HTS_Engine_set_gstream (HTS_Engine * *engine*, HTS_GStreamSet * *gss*)

A streaming engine method to set a different set of generated parameter stream.

Parameters

engine	a HTS_Engine pointer to be updated.
gss	a HTS_GStreamSet pointer to the new set of generated parameter stream.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.3.2.8 void p_HTS_Engine_set_gstream_pitch (HTS_Engine * *engine*, int *delay*, int * *total_frame_array*, double *pitch*, int *mode*)

A streaming engine method to set a different pitch value in the generated speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
delay	an integer for the number of labels used as delay, number of used future labels.
total_frame_array	a pointer to the array with the total number of frames.
pitch	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of pitch, if mode=1, the existing trajectory is shifted up if pitch>0 or shifted down if pitch<0.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.3.2.9 void p_HTS_Engine_set_labels (HTS_Engine * *engine*, HTS_Label * *label*)

A streaming engine method to set a different set of labels.

Parameters

engine	a HTS_Engine pointer to be updated.
label	a HTS_Label pointer to the new label set.

Attention

You must initialize the pHTS environment and the HTS engine before calling this

function.

4.3.2.10 void p_HTS_Engine_set_pstream (HTS_Engine * *engine*, HTS_PStreamSet * *pss*)

A streaming enfine method to set a different set of PDF streams.

Parameters

engine	a HTS_Engine pointer to be updated.
pss	a HTS_PStreamSet pointer to the new PDF stream set.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4 pHTS_lib/p_HTS_engine.h File Reference

Contains the decleration of all the streaming functions of the HTS_Engine.

```
#include "HTS_hidden.h"
```

Functions

- void p_HTS_ModelSet_get_duration (HTS_ModelSet *ms, char *label_str, double *mean, double *vari, double *iw)
- void p_HTS_Label_load_from_str (HTS_Label *label, int sampling_rate, int fperiod, char *label_str)
- void p_HTS_Label_load_from_fp (HTS_Label *label, int sampling_rate, int fperiod, FILE *fp)
- char * p_HTS_Label_get_string (HTS_Label *label)
- HTS_Boolean p_HTS_Label_get_frame_specified_flag (HTS_Label *label)
- int p_HTS_Label_get_frame (HTS_Label *label)
- void p_HTS_Label_set (HTS_Label *label, HTS_LabelString *head, HTS_LabelString *tail, int size, double speech_speed)
- void p_HTS_Label_set_frame (HTS_Label *label, int frame_length)
- void p_HTS_Label_set_speech_speed (HTS_Label *label, double speech_speed)
- void p_HTS_SStreamSet_initialize (HTS_SStreamSet *sss)
- void p_HTS_SStreamSet_create (HTS_SStreamSet *sss, HTS_ModelSet *ms, HTS_Label *label, double *duration_iw, double **parameter_iw, double **gv_iw, int **total_frame_array, float duration, int mode)
- void p_HTS_PStreamSet_create (HTS_PStreamSet *pss, HTS_SStreamSet *sss, double *msd_threshold, double *gv_weight, HTS_Label *label, HTS_ModelSet *ms, int *total_frame_array, int **length_array, int past)
- void p_HTS_PStreamSet_set (HTS_PStreamSet *pss, HTS_PStream *pstream, int nstream, int total_frame)

- void p_HTS_GStreamSet_initialize (HTS_GStreamSet *gss)
- void p_HTS_GStreamSet_create_param (HTS_GStreamSet *gss, HTS_PStreamSet *pss, HTS_SStreamSet *sss, HTS_Label *label, int stage, HTS_Boolean use_log_gain, int sampling_rate, int fperiod, double alpha, double beta, int audio_buff_size, int delay, int *total_frame_array, int *length_array, HTS_Boolean reset)
- void p_HTS_GStreamSet_create_speech_samples (HTS_GStreamSet *gss, HTS_PStreamSet *pss, HTS_SStreamSet *sss, HTS_Label *label, int stage, HTS_Boolean use_log_gain, int sampling_rate, int fperiod, double alpha, double beta, int audio_buff_size, int delay, HTS_Vocoder *v, int *total_frame_array)
- void p_HTS_GStreamSet_create (HTS_GStreamSet *gss, HTS_PStreamSet *pss, HTS_SStreamSet *sss, HTS_Label *label, int stage, HTS_Boolean use_log_gain, int sampling_rate, int fperiod, double alpha, double beta, int audio_buff_size, int delay, HTS_Vocoder *v, int *total_frame_array, int *length_array, HTS_Boolean reset)
- void p_HTS_GStreamSet_set_gstream (HTS_GStreamSet *gss, int total_nsample, int total_frame, int nstream, HTS_GStream *gstream, short *gspeech)
- void p_HTS_GStreamSet_set_gstream_pitch (HTS_GStreamSet *gss, HTS_SStreamSet *sss, HTS_Label *label, int delay, int *total_frame_array, double lf0, int mode)
- void p_HTS_Engine_load_label_from_str (HTS_Engine *engine, char *label_str)
- void p_HTS_Engine_load_label_from_fp (HTS_Engine *engine, FILE *fp)
- void p_HTS_Engine_create_sstream (HTS_Engine *engine, int **total_frame_array, double duration, int mode)
- void p_HTS_Engine_create_pstream (HTS_Engine *engine, int *total_frame_array, int **length_array, int past)
- void p_HTS_Engine_create_gstream_param (HTS_Engine *engine, int delay, int *total_frame_array, int *length_array, HTS_Boolean reset)
- void p_HTS_Engine_create_gstream_speech_samples (HTS_Engine *engine, int delay, HTS_Vocoder *v, int *total_frame_array)
- void p_HTS_Engine_set_labels (HTS_Engine *engine, HTS_Label *label)
- void p_HTS_Engine_set_pstream (HTS_Engine *engine, HTS_PStreamSet *pss)
- void p_HTS_Engine_set_gstream (HTS_Engine *engine, HTS_GStreamSet *gss)
- void p_HTS_Engine_set_gstream_pitch (HTS_Engine *engine, int delay, int *total_frame_array, double pitch, int mode)

4.4.1 Detailed Description

Contains the decleration of all the streaming functions of the HTS_Engine. The functions here, are the streaming version of the original HTS_Engine functions.

4.4.2 Function Documentation

4.4.2.1 void p_HTS_Engine_create_gstream_param (HTS_Engine * *engine*, int *delay*, int * *total_frame_array*, int * *length_array*, HTS_Boolean *reset*)

A streaming engine method to generate speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
delay	an integer for the number of labels used as delay, number of used future labels.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.
reset	a boolean integer for the number of labels to be synthesised before refreshing the engine. If reset=0 then the delayed frames are not estimated/synthesised, if reset=1 then the delayed framd are estimated/synthesised.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.
To synthesize speech, you must set stream[0]=spectrum models and spectrum[1]=lf0 models.

4.4.2.2 void p_HTS_Engine_create_gstream_speech_samples (HTS_Engine * *engine*, int *delay*, HTS_Vocoder * *v*, int * *total_frame_array*)

A streaming engine method to generate speech samples.

Parameters

engine	a HTS_Engine pointer to be updated.
delay	an integer for the number of labels used as delay, number of used future labels.
v	a HTS_Vocoder pointer to the structure for setting of vocoder.
total_frame_array	a pointer to the array with the total number of frames.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.
To synthesize speech, you must set stream[0]=spectrum models and spectrum[1]=lf0 models.

4.4.2.3 void p_HTS_Engine_create_pstream (HTS_Engine * *engine*, int * *total_frame_array*, int ** *length_array*, int *past*)

A streaming engine method to generate parameter using GV weight.

Parameters

engine	a HTS_Engine pointer to be updated.
total_frame_array	a pointer to the array with the total number of frames.

length_array	a pointer to the array with the stream lengths.
past	an integer for the number of past labels to be used.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.4 void p_HTS_Engine_create_sstream (HTS_Engine * *engine*, int ** *total_frame_array*, double *duration*, int *mode*)

A streaming engine method to parse label and determine state duration.

Parameters

engine	a HTS_Engine pointer to be updated.
total_frame_array	a pointer to the array with the total number of frames.
duration	a double for the duration of the phoneme in the label
mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.5 void p_HTS_Engine_load_label_from_fp (HTS_Engine * *engine*, FILE * *fp*)

A streaming engine method to load a phonetic label from file pointer.

Parameters

engine	a HTS_Engine pointer to be updated.
fp	a file pointer to the file containing phonetic labels.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.6 void p_HTS_Engine_load_label_from_str (HTS_Engine * *engine*, char * *label_str*)

A streaming engine method to load a phonetic label from a string.

Parameters

engine	a HTS_Engine pointer to be updated.
label_str	character pointer to a phonetic label.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.7 void p_HTS_Engine_set_gstream (HTS_Engine * *engine*, HTS_GStreamSet * *gss*)

A streaming engine method to set a different set of generated parameter stream.

Parameters

engine	a HTS_Engine pointer to be updated.
gss	a HTS_GStreamSet pointer to the new set of generated parameter stream.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.8 void p_HTS_Engine_set_gstream_pitch (HTS_Engine * *engine*, int *delay*, int * *total_frame_array*, double *pitch*, int *mode*)

A streaming engine method to set a different pitch value in the generated speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
delay	an integer for the number of labels used as delay, number of used future labels.
total_frame_array	a pointer to the array with the total number of frames.
pitch	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of pitch, if mode=1, the existing trajectory is shifted up if pitch>0 or shifted down if pitch<0.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.9 void p_HTS_Engine_set_labels (HTS_Engine * *engine*, HTS_Label * *label*)

A streaming engine method to set a different set of labels.

Parameters

engine	a HTS_Engine pointer to be updated.
label	a HTS_Label pointer to the new label set.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.10 void p_HTS_Engine_set_pstream (HTS_Engine * *engine*, HTS_PStreamSet * *pss*)

A streaming enfine method to set a different set of PDF streams.

Parameters

engine	a HTS_Engine pointer to be updated.
pss	a HTS_PStreamSet pointer to the new PDF stream set.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.11 void p_HTS_GStreamSet_create (HTS_GStreamSet * *gss*, HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *stage*, HTS_Boolean *use_log_gain*, int *sampling_rate*, int *fperiod*, double *alpha*, double *beta*, int *audio_buff_size*, int *delay*, HTS_Vocoder * *v*, int * *total_frame_array*, int * *length_array*, HTS_Boolean *reset*)

A streaming generated stream method to generate speech parameters and speech samples.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
pss	a HTS_PStreamSet pointer to the set of PDF streams.
sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
stage	an integer for stage, Gamma=-1, if stage=0 then Gamma=0.
use_log_gain	an integer for the log gain flag (for LSP).
sampling_rate	an integer for the sampling rate.
fperiod	an intefer for the frame period.
alpha	a double for the all-pass constant.

beta	a double for the postfiltering coefficient.
audio_buff_size	an integer for the audio buffer size (for audio device).
delay	an integer for the number of labels used as delay, number of used future labels.
reset	a boolean integer for the number of labels to be synthesised before refreshing the engine. If reset=0 then the delayed frames are not estimated/synthesised, if reset=1 then the delayed framd are estimated/synthesised.
v	a HTS_Vocoder pointer to the structure for setting of vocoder.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.12 void p_HTS_GStreamSet_create_param (HTS_GStreamSet * *gss*, HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *stage*, HTS_Boolean *use_log_gain*, int *sampling_rate*, int *fperiod*, double *alpha*, double *beta*, int *audio_buff_size*, int *delay*, int * *total_frame_array*, int * *length_array*, HTS_Boolean *reset*)

A streaming generated stream method to generate speech parameters.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
pss	a HTS_PStreamSet pointer to the set of PDF streams.
sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
stage	an integer for stage, Gamma=-1, if stage=0 then Gamma=0.
use_log_gain	an integer for the log gain flag (for LSP).
sampling_rate	an integer for the sampling rate.
fperiod	an intefer for the frame period.
alpha	a double for the all-pass constant.
beta	a double for the postfiltering coefficient.
audio_buff_size	an integer for the audio buffer size (for audio device).
delay	an integer for the number of labels used as delay, number of used future labels.
reset	a boolean integer for the number of labels to be synthesised before refreshing the engine. If reset=0 then the delayed frames are not estimated/synthesised, if reset=1 then the delayed framd are estimated/synthesised.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.13 void p_HTS_GStreamSet_create_speech_samples (HTS_GStreamSet * *gss*, HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *stage*, HTS_Boolean *use_log_gain*, int *sampling_rate*, int *fperiod*, double *alpha*, double *beta*, int *audio_buff_size*, int *delay*, HTS_Vocoder * *v*, int * *total_frame_array*)

A streaming generated stream method to generate speech samples.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
pss	a HTS_PStreamSet pointer to the set of PDF streams.
sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
stage	an integer for stage, Gamma=-1, if stage=0 then Gamma=0.
use_log_gain	an integer for the log gain flag (for LSP).
sampling_rate	an integer for the sampling rate.
fperiod	an intefer for the frame period.
alpha	a double for the all-pass constant.
beta	a double for the postfiltering coefficient.
audio_buff_size	an integer for the audio buffer size (for audio device).
delay	an integer for the number of labels used as delay, number of used future labels.
v	a HTS_Vocoder pointer to the structure for setting of vocoder.
total_frame_array	a pointer to the array with the total number of frames.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.14 void p_HTS_GStreamSet_initialize (HTS_GStreamSet * *gss*)

A streaming generated stream method to initialize the set of generated parameter stream.

Parameters

gss	a HTS_GStreamSet pointer to be initialized.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.15 void p_HTS_GStreamSet_set_gstream (HTS_GStreamSet * *gss*, int *total_nsample*, int *total_frame*, int *nstream*, HTS_GStream * *gstream*, short * *gspeech*)

A streaming generated stream method to set a different set of generated parameter stream.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
total_nsample	an integer for the number of samples.
total_frame	an integer for the total number of frames.
nstream	an integer for the number of stream.
gstream	a HTS_GStream pointer to the generated parameter stream.
gspeech	a short pointer to the generated speech samples.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.16 void p_HTS_GStreamSet_set_gstream_pitch (HTS_GStreamSet * *gss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *delay*, int * *total_frame_array*, double *lf0*, int *mode*)

A streaming generated stream method to set a different pitch value in the generated speech parameters.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
delay	an integer for the number of labels used as delay, number of used future labels.
total_frame_array	a pointer to the array with the total number of frames.
lf0	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of lf0, if mode=1, the existing trajectory is shifted up if lf0$>$0 or shifted down if lf0$<$0.

Attention

You must initialize the pHTS environment and the HTS engine before calling this

function.

4.4.2.17 int p_HTS_Label_get_frame (HTS_Label * *label*)

A streaming label method to get the frame length from the last element of the label set.

Parameters

label	a HTS_Label pointer to be accessed.

Returns

an integer for the frame length.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.18 HTS_Boolean p_HTS_Label_get_frame_specified_flag (HTS_Label * *label*)

A streaming label method to get the frame specified flag from the last element of the label set.

Parameters

label	a HTS_Label pointer to be accessed.

Returns

an integer for the frame specified flag, 0 for false or 1 for true.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.19 char* p_HTS_Label_get_string (HTS_Label * *label*)

A streaming label method to access the phonetic label string from the last element of the label set.

Parameters

label	a HTS_Label pointer to be accessed.

Returns

a character pointer to the phonetic label string.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.20 void p_HTS_Label_load_from_fp (HTS_Label * *label*, int *sampling_rate*, int *fperiod*, FILE * *fp*)

A streaming label method to load a phonetic label from file pointer.

Parameters

label	a HTS_Label pointer to be updated.
sampling_rate	an integer for the sampling rate.
fperiod	an integer for the frame period.
fp	a file pointer to the file containing phonetic labels.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.21 void p_HTS_Label_load_from_str (HTS_Label * *label*, int *sampling_rate*, int *fperiod*, char * *label_str*)

A streaming label method to load a phonetic label from a string.

Parameters

label	a HTS_Label pointer to be updated.
sampling_rate	an integer for the sampling rate.
fperiod	an integer for the frame period.
label_str	a character pointer to a phonetic label.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.22 void p_HTS_Label_set (HTS_Label * *label*, HTS_LabelString * *head*, HTS_LabelString * *tail*, int *size*, double *speech_speed*)

A streaming label method to set a different set of labels.

Parameters

label	a HTS_Label pointer to be updated.
head	a HTS_LabelString pointer to the head pointer of the HTS_LabelString list.
tail	a HTS_LabelString pointer to the tail pointer of the HTS_LabelString list.
size	an integer for the size of the list, number of elements.
speech_speed	a double for the speech speed rate.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.23 void p_HTS_Label_set_frame (HTS_Label * *label*, int *frame_length*)

A streaming label method to set the frame length of the last element of the label set.

Parameters

label	a HTS_Label pointer to be updated.
frame_length	an integer for the frame length.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.24 void p_HTS_Label_set_speech_speed (HTS_Label * *label*, double *speech_speed*)

A streaming label method to set the speech speed rate of the last element of the label set.

Parameters

label	a HTS_Label pointer to be updated.
speech_speed	a double for the speech speed rate.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.25 void p_HTS_ModelSet_get_duration (HTS_ModelSet * *ms*, char * *label_str*, double * *mean*, double * *vari*, double * *iw*)

A streaming model method to get the duration using interpolation weight.

Parameters

ms	a HTS_ModelSet pointer to be updated.
label_str	a character pointer to a phonetic label.
mean	a pointer to a mean vector sequence.
vari	a pointer to a variance vector sequence.
iw	a pointer to the weights for duration interpolation.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.26 void p_HTS_PStreamSet_create (HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, double * *msd_threshold*, double * *gv_weight*, HTS_Label * *label*, HTS_ModelSet * *ms*, int * *total_frame_array*, int ** *length_array*, int *past*)

A streaming PDF stream method to generate parameter using GV weight.

Parameters

pss	a HTS_PStreamSet pointer to be updated.
sss	a HTS_SStreamSet pointer to the set of state streams.
msd_threshold	a pointer to the MSD thresholds.
gv_weight	a pointer to the GV weight.
label	a pointer to the label set.
ms	a pointer to the set of duration models, HMMs and GV models.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.
past	an integer for the number of past labels to be used.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.27 void p_HTS_PStreamSet_set (HTS_PStreamSet * *pss*, HTS_PStream * *pstream*, int *nstream*, int *total_frame*)

A streaming PDF stream method to set a different set of PDF streams.

Parameters

pss	a HTS_PStreamSet pointer to be updated.
pstream	a HTS_PStream pointer to the PDF streams.
nstream	an integer for the number of PDF streams.
total_frame	an integer for the total number of frames.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.28 void p_HTS_SStreamSet_create (HTS_SStreamSet * *sss*, HTS_ModelSet * *ms*, HTS_Label * *label*, double * *duration_iw*, double ** *parameter_iw*, double ** *gv_iw*, int ** *total_frame_array*, float *duration*, int *mode*)

A streaming state stream method to parse label and determine state duration.

Parameters

sss	a HTS_SStreamSet pointer to be updated.
ms	a HTS_ModelSet pointer to the set of duration models, HMMs and GV models.
label	a HTS_Label pointer to the label set.
duration_iw	a pointer to the weights for duration interpolation.
parameter_iw	a pointer to the weights for parameter interpolation.
gv_iw	a pointer to the weights for GV interpolation.
total_frame_array	a pointer to the array with the total number of frames.
duration	a double for the duration of the phoneme in the label
mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.4.2.29 void p_HTS_SStreamSet_initialize (HTS_SStreamSet * *sss*)

A streaming state stream method to initialize the state stream set.

Parameters

sss	a HTS_SStreamSet pointer to be initialized.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.5 pHTS_lib/p_HTS_gstream.c File Reference

Contains the implementation of all the streaming functions of the HTS_Engine for the set of the generated parameter stream.

```
#include <stdlib.h>
#include "HTS_hidden.h"
```

Functions

- void p_HTS_GStreamSet_initialize (HTS_GStreamSet *gss)
- void p_HTS_GStreamSet_create_param (HTS_GStreamSet *gss, HTS_PStreamSet *pss, HTS_SStreamSet *sss, HTS_Label *label, int stage, HTS_Boolean use_log_gain, int sampling_rate, int fperiod, double alpha, double beta, int audio_buff_size, int delay, int *total_frame_array, int *length_array, HTS_Boolean reset)
- void p_HTS_GStreamSet_create_speech_samples (HTS_GStreamSet *gss, HTS_PStreamSet *pss, HTS_SStreamSet *sss, HTS_Label *label, int stage, HTS_Boolean use_log_gain, int sampling_rate, int fperiod, double alpha, double beta, int audio_buff_size, int delay, HTS_Vocoder *v, int *total_frame_array)
- void p_HTS_GStreamSet_create (HTS_GStreamSet *gss, HTS_PStreamSet *pss, HTS_SStreamSet *sss, HTS_Label *label, int stage, HTS_Boolean use_log_gain, int sampling_rate, int fperiod, double alpha, double beta, int audio_buff_size, int delay, HTS_Vocoder *v, int *total_frame_array, int *length_array, HTS_Boolean reset)
- void p_HTS_GStreamSet_set_gstream (HTS_GStreamSet *gss, int total_nsample, int total_frame, int nstream, HTS_GStream *gstream, short *gspeech)
- void p_HTS_GStreamSet_set_gstream_pitch (HTS_GStreamSet *gss, HTS_SStreamSet *sss, HTS_Label *label, int delay, int *total_frame_array, double lf0, int mode)

4.5.1 Detailed Description

Contains the implementation of all the streaming functions of the HTS_Engine for the set of the generated parameter stream. The functions implemented here, are the streaming version of the original gsteam functions of the HTS_Engine.

4.5.2 Function Documentation

4.5.2.1 void p_HTS_GStreamSet_create (HTS_GStreamSet * *gss*, HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *stage*, HTS_Boolean *use_log_gain*, int *sampling_rate*, int *fperiod*, double *alpha*, double *beta*, int *audio_buff_size*, int *delay*, HTS_Vocoder * *v*, int * *total_frame_array*, int * *length_array*, HTS_Boolean *reset*)

A streaming generated stream method to generate speech parameters and speech samples.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
pss	a HTS_PStreamSet pointer to the set of PDF streams.
sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
stage	an integer for stage, Gamma=-1, if stage=0 then Gamma=0.
use_log_gain	an integer for the log gain flag (for LSP).
sampling_rate	an integer for the sampling rate.
fperiod	an intefer for the frame period.
alpha	a double for the all-pass constant.
beta	a double for the postfiltering coefficient.
audio_buff_size	an integer for the audio buffer size (for audio device).
delay	an integer for the number of labels used as delay, number of used future labels.
reset	a boolean integer for the number of labels to be synthesised before refreshing the engine. If reset=0 then the delayed frames are not estimated/synthesised, if reset=1 then the delayed framd are estimated/synthesised.
v	a HTS_Vocoder pointer to the structure for setting of vocoder.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.5.2.2 void p_HTS_GStreamSet_create_param (HTS_GStreamSet * *gss*, HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *stage*, HTS_Boolean *use_log_gain*, int *sampling_rate*, int *fperiod*, double *alpha*, double *beta*, int *audio_buff_size*, int *delay*, int * *total_frame_array*, int * *length_array*, HTS_Boolean *reset*)

A streaming generated stream method to generate speech parameters.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
pss	a HTS_PStreamSet pointer to the set of PDF streams.
sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
stage	an integer for stage, Gamma=-1, if stage=0 then Gamma=0.
use_log_gain	an integer for the log gain flag (for LSP).
sampling_rate	an integer for the sampling rate.

fperiod	an intefer for the frame period.
alpha	a double for the all-pass constant.
beta	a double for the postfiltering coefficient.
audio_buff_size	an integer for the audio buffer size (for audio device).
delay	an integer for the number of labels used as delay, number of used future labels.
reset	a boolean integer for the number of labels to be synthesised before refreshing the engine. If reset=0 then the delayed frames are not estimated/synthesised, if reset=1 then the delayed framd are estimated/synthesised.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.5.2.3 void p_HTS_GStreamSet_create_speech_samples (HTS_GStreamSet * *gss*, HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *stage*, HTS_Boolean *use_log_gain*, int *sampling_rate*, int *fperiod*, double *alpha*, double *beta*, int *audio_buff_size*, int *delay*, HTS_Vocoder * *v*, int * *total_frame_array*)

A streaming generated stream method to generate speech samples.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
pss	a HTS_PStreamSet pointer to the set of PDF streams.
sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
stage	an integer for stage, Gamma=-1, if stage=0 then Gamma=0.
use_log_gain	an integer for the log gain flag (for LSP).
sampling_rate	an integer for the sampling rate.
fperiod	an intefer for the frame period.
alpha	a double for the all-pass constant.
beta	a double for the postfiltering coefficient.
audio_buff_size	an integer for the audio buffer size (for audio device).
delay	an integer for the number of labels used as delay, number of used future labels.
v	a HTS_Vocoder pointer to the structure for setting of vocoder.
total_frame_array	a pointer to the array with the total number of frames.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.5.2.4 void p_HTS_GStreamSet_initialize (HTS_GStreamSet * *gss*)

A streaming generated stream method to initialize the set of generated parameter stream.

Parameters

gss	a HTS_GStreamSet pointer to be initialized.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.5.2.5 void p_HTS_GStreamSet_set_gstream (HTS_GStreamSet * *gss*, int *total_nsample*, int *total_frame*, int *nstream*, HTS_GStream * *gstream*, short * *gspeech*)

A streaming generated stream method to set a different set of generated parameter stream.

Parameters

gss	a HTS_GStreamSet pointer to be updated.
total_nsample	an integer for the number of samples.
total_frame	an integer for the total number of frames.
nstream	an integer for the number of stream.
gstream	a HTS_GStream pointer to the generated parameter stream.
gspeech	a short pointer to the generated speech samples.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.5.2.6 void p_HTS_GStreamSet_set_gstream_pitch (HTS_GStreamSet * *gss*, HTS_SStreamSet * *sss*, HTS_Label * *label*, int *delay*, int * *total_frame_array*, double *lf0*, int *mode*)

A streaming generated stream method to set a different pitch value in the generated speech parameters.

Parameters

gss	a HTS_GStreamSet pointer to be updated.

sss	a HTS_SStreamSet pointer to the set of state stream.
label	a HTS_Label pointer to the label set.
delay	an integer for the number of labels used as delay, number of used future labels.
total_frame_array	a pointer to the array with the total number of frames.
lf0	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of lf0, if mode=1, the existing trajectory is shifted up if lf0>0 or shifted down if lf0<0.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6 pHTS_lib/p_HTS_label.c File Reference

Contains the implementation of all the streaming functions of the HTS_Engine for the labels set.

```
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include "HTS_hidden.h"
```

Functions

- void p_HTS_Label_load_from_str (HTS_Label *label, int sampling_rate, int fperiod, char *label_str)
- void p_HTS_Label_load_from_fp (HTS_Label *label, int sampling_rate, int fperiod, FILE *fp)
- char * p_HTS_Label_get_string (HTS_Label *label)
- HTS_Boolean p_HTS_Label_get_frame_specified_flag (HTS_Label *label)
- int p_HTS_Label_get_frame (HTS_Label *label)
- void p_HTS_Label_set (HTS_Label *label, HTS_LabelString *head, HTS_LabelString *tail, int size, double speech_speed)
- void p_HTS_Label_set_frame (HTS_Label *label, int frame_length)
- void p_HTS_Label_set_speech_speed (HTS_Label *label, double speech_speed)

4.6.1 Detailed Description

Contains the implementation of all the streaming functions of the HTS_Engine for the labels set. The functions implemented here, are the streaming version of the original label functions of the HTS_Engine.

4.6.2 Function Documentation

4.6.2.1 int p_HTS_Label_get_frame (HTS_Label * *label*)

A streaming label method to get the frame length from the last element of the label set.

Parameters

label	a HTS_Label pointer to be accessed.

Returns

an integer for the frame length.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6.2.2 HTS_Boolean p_HTS_Label_get_frame_specified_flag (HTS_Label * *label*)

A streaming label method to get the frame specified flag from the last element of the label set.

Parameters

label	a HTS_Label pointer to be accessed.

Returns

an integer for the frame specified flag, 0 for false or 1 for true.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6.2.3 char* p_HTS_Label_get_string (HTS_Label * *label*)

A streaming label method to access the phonetic label string from the last element of the label set.

Parameters

label	a HTS_Label pointer to be accessed.

Returns

a character pointer to the phonetic label string.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6.2.4 void p_HTS_Label_load_from_fp (HTS_Label * *label*, int *sampling_rate*, int *fperiod*, FILE * *fp*)

A streaming label method to load a phonetic label from file pointer.

Parameters

label	a HTS_Label pointer to be updated.
sampling_rate	an integer for the sampling rate.
fperiod	an integer for the frame period.
fp	a file pointer to the file containing phonetic labels.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6.2.5 void p_HTS_Label_load_from_str (HTS_Label * *label*, int *sampling_rate*, int *fperiod*, char * *label_str*)

A streaming label method to load a phonetic label from a string.

Parameters

label	a HTS_Label pointer to be updated.
sampling_rate	an integer for the sampling rate.
fperiod	an integer for the frame period.
label_str	a character pointer to a phonetic label.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6.2.6 void p_HTS_Label_set (HTS_Label * *label*, HTS_LabelString * *head*, HTS_LabelString * *tail*, int *size*, double *speech_speed*)

A streaming label method to set a different set of labels.

Parameters

label	a HTS_Label pointer to be updated.
head	a HTS_LabelString pointer to the head pointer of the HTS_LabelString list.
tail	a HTS_LabelString pointer to the tail pointer of the HTS_LabelString list.
size	an integer for the size of the list, number of elements.
speech_speed	a double for the speech speed rate.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6.2.7 void p_HTS_Label_set_frame (HTS_Label * *label*, int *frame_length*)

A streaming label method to set the frame length of the last element of the label set.

Parameters

label	a HTS_Label pointer to be updated.
frame_length	an integer for the frame length.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.6.2.8 void p_HTS_Label_set_speech_speed (HTS_Label * *label*, double *speech_speed*)

A streaming label method to set the speech speed rate of the last element of the label set.

Parameters

label	a HTS_Label pointer to be updated.
speech_speed	a double for the speech speed rate.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.7 pHTS_lib/p_HTS_model.c File Reference

Contains the implementation of all the streaming functions of the HTS_Engine for the set of duration models, HMMs and GV models.

```
#include <string.h>
#include "HTS_hidden.h"
```

Functions

- void p_HTS_ModelSet_get_duration (HTS_ModelSet *ms, char *label_str, double *mean, double *vari, double *iw)

4.7.1 Detailed Description

Contains the implementation of all the streaming functions of the HTS_Engine for the set of duration models, HMMs and GV models. The functions implemented here, are the streaming version of the original model functions of the HTS_Engine.

4.7.2 Function Documentation

4.7.2.1 void p_HTS_ModelSet_get_duration (HTS_ModelSet * *ms*, char * *label_str*, double * *mean*, double * *vari*, double * *iw*)

A streaming model method to get the duration using interpolation weight.

Parameters

ms	a HTS_ModelSet pointer to be updated.
label_str	a character pointer to a phonetic label.
mean	a pointer to a mean vector sequence.
vari	a pointer to a variance vector sequence.
iw	a pointer to the weights for duration interpolation.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.8 pHTS_lib/p_HTS_pstream.c File Reference

Contains the implementation of all the streaming functions of the HTS_Engine for the set of the PDF streams.

```
#include <math.h>
#include <stdlib.h>
```

```
#include "HTS_hidden.h"
```

Functions

- void p_HTS_PStreamSet_create (HTS_PStreamSet *pss, HTS_SStreamSet *sss, double *msd_threshold, double *gv_weight, HTS_Label *label, HTS_ModelSet *ms, int *total_frame_array, int **length_array, int past)
- void p_HTS_PStreamSet_set (HTS_PStreamSet *pss, HTS_PStream *pstream, int nstream, int total_frame)

4.8.1 Detailed Description

Contains the implementation of all the streaming functions of the HTS_Engine for the set of the PDF streams. The functions implemented here, are the streaming version of the original pstream functions of the HTS_Engine.

4.8.2 Function Documentation

4.8.2.1 void p_HTS_PStreamSet_create (HTS_PStreamSet * *pss*, HTS_SStreamSet * *sss*, double * *msd_threshold*, double * *gv_weight*, HTS_Label * *label*, HTS_ModelSet * *ms*, int * *total_frame_array*, int ** *length_array*, int *past*)

A streaming PDF stream method to generate parameter using GV weight.

Parameters

pss	a HTS_PStreamSet pointer to be updated.
sss	a HTS_SStreamSet pointer to the set of state streams.
msd_threshold	a pointer to the MSD thresholds.
gv_weight	a pointer to the GV weight.
label	a pointer to the label set.
ms	a pointer to the set of duration models, HMMs and GV models.
total_frame_array	a pointer to the array with the total number of frames.
length_array	a pointer to the array with the stream lengths.
past	an integer for the number of past labels to be used.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.8.2.2 void p_HTS_PStreamSet_set (HTS_PStreamSet * *pss*, HTS_PStream * *pstream*, int *nstream*, int *total_frame*)

A streaming PDF stream method to set a different set of PDF streams.

Parameters

pss	a HTS_PStreamSet pointer to be updated.
pstream	a HTS_PStream pointer to the PDF streams.
nstream	an integer for the number of PDF streams.
total_frame	an integer for the total number of frames.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.9 pHTS_lib/p_HTS_sstream.c File Reference

Contains the implementation of all the streaming functions of the HTS_Engine for the set of state stream.

```
#include <stdlib.h>
#include "HTS_hidden.h"
```

Functions

- void p_HTS_SStreamSet_initialize (HTS_SStreamSet *sss)
- void p_HTS_SStreamSet_create (HTS_SStreamSet *sss, HTS_ModelSet *ms, HTS_Label *label, double *duration_iw, double **parameter_iw, double **gv_iw, int **total_frame_array, float duration, int mode)

4.9.1 Detailed Description

Contains the implementation of all the streaming functions of the HTS_Engine for the set of state stream. The functions implemented here, are the streaming version of the original ssteam functions of the HTS_Engine.

4.9.2 Function Documentation

4.9.2.1 void p_HTS_SStreamSet_create (HTS_SStreamSet * *sss*, HTS_ModelSet * *ms*, HTS_Label * *label*, double * *duration_iw*, double ** *parameter_iw*, double ** *gv_iw*, int ** *total_frame_array*, float *duration*, int *mode*)

A streaming state stream method to parse label and determine state duration.

Parameters

sss	a HTS_SStreamSet pointer to be updated.
ms	a HTS_ModelSet pointer to the set of duration models, HMMs and GV models.
label	a HTS_Label pointer to the label set.

duration_iw	a pointer to the weights for duration interpolation.
parameter_iw	a pointer to the weights for parameter interpolation.
gv_iw	a pointer to the weights for GV interpolation.
total_frame_array	a pointer to the array with the total number of frames.
duration	a double for the duration of the phoneme in the label
mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.9.2.2 void p_HTS_SStreamSet_initialize (HTS_SStreamSet * *sss*)

A streaming state stream method to initialize the state stream set.

Parameters

sss	a HTS_SStreamSet pointer to be initialized.

Attention

You must initialize the pHTS environment and the HTS engine before calling this function.

4.10 pHTS_lib/pHTS.c File Reference

Contains the implementation of all the streaming functions of the streaming pHTS environment.

```
#include "pHTS.h"
#include <unistd.h>
```

Functions

- void pHTS_initialize (int num_arg, char **arguments, HTS_Engine *engine, pHTS *p)
- void pHTS_pushLabel (HTS_Engine *engine, pHTS *p, char *label)
- int pHTS_parseLabel (HTS_Engine *engine, pHTS *p)
- void pHTS_updateEngine (HTS_Engine *engine, pHTS *p)
- void pHTS_updatePDFs (HTS_Engine *engine, pHTS *p)

- void pHTS_updateFilter (HTS_Engine *engine, pHTS *p)
- void pHTS_updateSamples (HTS_Engine *engine, pHTS *p)
- int pHTS_getNumberOfLabels (HTS_Engine *engine, pHTS *p)
- int pHTS_getNumberOfSamples (HTS_Engine *engine, pHTS *p)
- void pHTS_getSamples (HTS_Engine *engine, pHTS *p, short *buffer, int nOfSamples)
- void pHTS_popSamples (HTS_Engine *engine, pHTS *p, float *buffer, int nOfSamples)
- HTS_LabelString * pHTS_getLabel (HTS_Engine *engine, pHTS *p)
- HTS_PStreamSet * pHTS_getPDFs (HTS_Engine *engine, pHTS *p)
- HTS_GStreamSet * pHTS_getFilter (HTS_Engine *engine, pHTS *p)
- double pHTS_getSpeed (HTS_Engine *engine, pHTS *p)
- double pHTS_getPitch (HTS_Engine *engine, pHTS *p)
- double pHTS_getVolume (HTS_Engine *engine, pHTS *p)
- double pHTS_getAlpha (HTS_Engine *engine, pHTS *p)
- double pHTS_getDuration (HTS_Engine *engine, pHTS *p)
- void pHTS_setLabels (HTS_Engine *engine, pHTS *p, HTS_Label *label)
- void pHTS_setPDFs (HTS_Engine *engine, pHTS *p, HTS_PStreamSet *pss)
- void pHTS_setFilter (HTS_Engine *engine, pHTS *p, HTS_GStreamSet *gss)
- void pHTS_setSpeed (HTS_Engine *engine, pHTS *p, double speechSpeed)
- void pHTS_setPitch (HTS_Engine *engine, pHTS *p, double pitch, int mode)
- void pHTS_setVolume (HTS_Engine *engine, pHTS *p, double volume)
- void pHTS_setAlpha (HTS_Engine *engine, pHTS *p, double alpha)
- void pHTS_setDuration (HTS_Engine *engine, pHTS *p, double duration, int mode)
- char * pHTS_getLabelFromFile (HTS_Engine *engine, pHTS *p)
- void pHTS_finalize (HTS_Engine *engine, pHTS *p)
- void pHTS_refresh (HTS_Engine *engine, pHTS *p)
- void pHTS_free (HTS_Engine *engine, pHTS *p)

4.10.1 Detailed Description

Contains the implementation of all the streaming functions of the streaming pHTS environment.

4.10.2 Function Documentation

4.10.2.1 void pHTS_finalize (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to flush the delayed labels.

Parameters

engine	a HTS_Engine pointer to be accessed and refreshed.
p	a pHTS pointer to be accessed.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.10.2.2 void pHTS_free (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to free all the alocated memory.

Parameters

engine	a HTS_Engine pointer to be freed.
p	a pHTS pointer to be accessed.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.10.2.3 double pHTS_getAlpha (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the alpha value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the alpha, all-pass constant value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.4 double pHTS_getDuration (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the duration value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the duration value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.5 HTS_GStreamSet* pHTS_getFilter (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the set of generated parameter stream.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a HTS_GStreamSet pointer to the set of generated parameter stream.

Attention

You must generate the speech parameters from the PDFs of the last parsed label in the pHTS environment using pHTS_updateFilter before calling this function.

4.10.2.6 HTS_LabelString* pHTS_getLabel (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the last label from the label set.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a HTS_Label pointer to the last label from the label set.

Attention

You must push a label in the pHTS environment using pHTS_pushLabel before calling this function.

4.10.2.7 char* pHTS_getLabelFromFile (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get one label string from a label file.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a pointer to the new label string retrieved from the label file.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.10.2.8 int pHTS_getNumberOfLabels (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the number of labels from a label file.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

an integer for the number of labels from a label file.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.9 int pHTS_getNumberOfSamples (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the number of the generated samples.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

an integer for the number of the generated samples.

Attention

You must generate the speech samples of the last parsed label in the pHTS environment using pHTS_updateSamples before calling this function.

4.10.2.10 HTS_PStreamSet* pHTS_getPDFs (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the set of PDF streams.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a HTS_Label pointer to the PDF stream set.

Attention

You must generate the PDFs of the last parsed label in the pHTS environment using pHTS_updatePDFs before calling this function.

4.10.2.11 double pHTS_getPitch (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the pitch value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the pitch value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.12 void pHTS_getSamples (HTS_Engine * *engine*, pHTS * *p*, short * *buffer*, int *nOfSamples*)

A streaming environment method to get the generated speech samples.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.
buffer	short pointer to the buffer to be filled with the generated samples.
nOfSamples	an integer for the number of samples to be copied in in the buffer.

Attention

You must generate the speech samples of the last parsed label in the pHTS environment using pHTS_updateSamples before calling this function.

4.10.2.13 double pHTS_getSpeed (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the speech speed rate.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the speech speed rate.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.14 double pHTS_getVolume (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the volume value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the volume value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.15 void pHTS_initialize (int *num_arg*, char ** *arguments*, HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to initialize all environment variables.

Parameters

num_arg	an integer for the number of arguments (argc).
arguments	a character pointer to all arguments (argv).
engine	a HTS_Engine pointer to be initialized.
p	a pHTS pointer to be initialized.

Attention

To start the pHTS environment, first you must call this function.

4.10.2.16 int pHTS_parseLabel (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to parse the last inserted label.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Returns

an integer, 0 if the label ring buffer is full, and -1 is there is still space.

Attention

You must push a label in the pHTS environment using pHTS_pushLabel before calling this function.

4.10.2.17 void pHTS_popSamples (HTS_Engine * *engine*, pHTS * *p*, float * *buffer*, int *nOfSamples*)

A streaming environment method to access and pop the generated speech samples from ring buffer.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.
buffer	float pointer to the buffer to be filled with the generated samples.
nOfSamples	an integer for the number of samples to be poped out of the ring buffer.

Attention

Must be called from audio callback

4.10.2.18 void pHTS_pushLabel (HTS_Engine * *engine*, pHTS * *p*, char * *label*)

A streaming environment method to set a new label into the labels set in order to be prosessed.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
label	character pointer to a phonetic label.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.19 void pHTS_refresh (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to flush the internal memory of the engine occupied by the pHTS environment.

Parameters

engine	a HTS_Engine pointer to be accessed and refreshed.
p	a pHTS pointer to be accessed.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.10.2.20 void pHTS_setAlpha (HTS_Engine * *engine*, pHTS * *p*, double *alpha*)

A streaming environment method to set different alpha value, the all-pass constant.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
alpha	a double for the new alpha, all-pass constant value.

Attention

Values less than zero or greater then one are ignored. Default value is 0.42.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.21 void pHTS_setDuration (HTS_Engine * *engine*, pHTS * *p*, double *duration*, int *mode*)

A streaming environment method to set different duration value in the generated speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
duration	a double for the new duration value.

mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

Attention

Values less than zero are ignored if mode=0 or mode=2.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.22 void pHTS_setFilter (HTS_Engine * *engine*, pHTS * *p*, HTS_GStreamSet * *gss*)

A streaming environment method to set a new set of generated parameter stream.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
gss	a HTS_GStreamSet pointer to the new set of generated parameter stream.

Attention

You must initialize the pHTS environment using pHTS_initialize and also create a valid set of generated parameters before calling this function.

4.10.2.23 void pHTS_setLabels (HTS_Engine * *engine*, pHTS * *p*, HTS_Label * *label*)

A streaming environment method to set a new set of labels.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
label	a HTS_Label pointer to the new label set.

Attention

You must initialize the pHTS environment using pHTS_initialize and also create a valid set of labels before calling this function.

4.10.2.24 void pHTS_setPDFs (HTS_Engine * *engine*, pHTS * *p*, HTS_PStreamSet * *pss*)

A streaming environment method to set a new set of PDF streams.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
pss	a HTS_PStreamSet pointer to the new PDF streams set.

Attention

You must initialize the pHTS environment using pHTS_initialize and also create a valid set of PDFs before calling this function.

4.10.2.25 void pHTS_setPitch (HTS_Engine * *engine*, pHTS * *p*, double *pitch*, int *mode*)

A streaming environment method to set different pitch value in the generated speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
pitch	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of pitch, if mode=1, the existing trajectory is shifted up if pitch>0 or shifted down if pitch<0.

Attention

the engine is using log f0.
values less than zero are ignored if mode=0.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.26 void pHTS_setSpeed (HTS_Engine * *engine*, pHTS * *p*, double *speechSpeed*)

A streaming environment method to set different speech speed rate.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
speech-Speed	a double for the new speech speed rate.

Attention

Values less than zero are ignored. Default value is 1.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.27 void pHTS_setVolume (HTS_Engine * *engine*, pHTS * *p*, double *volume*)

A streaming environment method to set different volume.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
volume	a double for the new volume.

Attention

Values less than zero are ignored. Default value is 1.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.10.2.28 void pHTS_updateEngine (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the speech samples from PDFs.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must parse the last pushed label in the pHTS environment using pHTS_parseLabel before calling this function.

4.10.2.29 void pHTS_updateFilter (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the speech parameters from the PDFs.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must generate the PDFs of the last parsed label in the pHTS environment using pHTS_updatePDFs before calling this function.

4.10.2.30 void pHTS_updatePDFs (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the PDFs.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must parse the last pushed label in the pHTS environment using pHTS_parseLabel before calling this function.

4.10.2.31 void pHTS_updateSamples (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the speech samples from the speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must generate the speech parameters from the PDFs of the last parsed label in the pHTS environment using pHTS_updateFilter before calling this function.

4.11 pHTS_lib/pHTS.h File Reference

Contains the decleration of all the streaming functions of the streaming pHTS environment.

```
#include <math.h>
#include <stdlib.h>
#include <stdarg.h>
#include "HTS_engine.h"
#include "p_HTS_engine.h"
#include "RingBuffer.h"
```

Classes

- struct _pHTS

Defines

- #define LABEL_BUFFER_SIZE 32

  A macro that defines the number of labels kept in the label ring buffer.

- #define PHTS_AUDIO_RINGBUFFER_SIZE 4096

  A macro that defines the number of samples kept in the audio ring buffer.

Typedefs

- typedef struct _pHTS pHTS

  A type definition for the pHTS environment.

Functions

- void pHTS_initialize (int num_arg, char **arguments, HTS_Engine *engine, pHTS *p)
- void pHTS_pushLabel (HTS_Engine *engine, pHTS *p, char *label)
- int pHTS_parseLabel (HTS_Engine *engine, pHTS *p)
- void pHTS_updateEngine (HTS_Engine *engine, pHTS *p)
- void pHTS_updatePDFs (HTS_Engine *engine, pHTS *p)
- void pHTS_updateFilter (HTS_Engine *engine, pHTS *p)
- void pHTS_updateSamples (HTS_Engine *engine, pHTS *p)
- int pHTS_getNumberOfLabels (HTS_Engine *engine, pHTS *p)
- int pHTS_getNumberOfSamples (HTS_Engine *engine, pHTS *p)
- void pHTS_getSamples (HTS_Engine *engine, pHTS *p, short *buffer, int nOfSamples)
- void pHTS_popSamples (HTS_Engine *engine, pHTS *p, float *buffer, int nOfSamples)
- HTS_LabelString * pHTS_getLabel (HTS_Engine *engine, pHTS *p)
- HTS_PStreamSet * pHTS_getPDFs (HTS_Engine *engine, pHTS *p)
- HTS_GStreamSet * pHTS_getFilter (HTS_Engine *engine, pHTS *p)
- double pHTS_getSpeed (HTS_Engine *engine, pHTS *p)
- double pHTS_getPitch (HTS_Engine *engine, pHTS *p)
- double pHTS_getVolume (HTS_Engine *engine, pHTS *p)
- double pHTS_getAlpha (HTS_Engine *engine, pHTS *p)
- double pHTS_getDuration (HTS_Engine *engine, pHTS *p)
- void pHTS_setLabels (HTS_Engine *engine, pHTS *p, HTS_Label *label)
- void pHTS_setPDFs (HTS_Engine *engine, pHTS *p, HTS_PStreamSet *pss)
- void pHTS_setFilter (HTS_Engine *engine, pHTS *p, HTS_GStreamSet *gss)
- void pHTS_setSpeed (HTS_Engine *engine, pHTS *p, double speechSpeed)
- void pHTS_setPitch (HTS_Engine *engine, pHTS *p, double pitch, int mode)
- void pHTS_setVolume (HTS_Engine *engine, pHTS *p, double volume)
- void pHTS_setAlpha (HTS_Engine *engine, pHTS *p, double alpha)
- void pHTS_setDuration (HTS_Engine *engine, pHTS *p, double duration, int mode)
- char * pHTS_getLabelFromFile (HTS_Engine *engine, pHTS *p)
- void pHTS_finalize (HTS_Engine *engine, pHTS *p)
- void pHTS_refresh (HTS_Engine *engine, pHTS *p)
- void pHTS_free (HTS_Engine *engine, pHTS *p)

4.11.1 Detailed Description

Contains the decleration of all the streaming functions of the streaming pHTS environment. The functions here, are the streaming version of the original HTS_Engine functions combined with the streaming pHTS environment.

4.11.2 Typedef Documentation

4.11.2.1 typedef struct _pHTS pHTS

A type definition for the pHTS environment.

The pHTS struct contains all the passed arguments for the engine, as well as the vocoder, and the variables needed for the streaming processes.

4.11.3 Function Documentation

4.11.3.1 void pHTS_finalize (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to flush the delayed labels.

Parameters

engine	a HTS_Engine pointer to be accessed and refreshed.
p	a pHTS pointer to be accessed.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.11.3.2 void pHTS_free (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to free all the alocated memory.

Parameters

engine	a HTS_Engine pointer to be freed.
p	a pHTS pointer to be accessed.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.11.3.3 double pHTS_getAlpha (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the alpha value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the alpha, all-pass constant value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.4 double pHTS_getDuration (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the duration value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the duration value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.5 HTS_GStreamSet* pHTS_getFilter (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the set of generated parameter stream.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a HTS_GStreamSet pointer to the set of generated parameter stream.

Attention

You must generate the speech parameters from the PDFs of the last parsed label in the pHTS environment using pHTS_updateFilter before calling this function.

4.11.3.6 HTS_LabelString* pHTS_getLabel (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the last label from the label set.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a HTS_Label pointer to the last label from the label set.

Attention

You must push a label in the pHTS environment using pHTS_pushLabel before calling this function.

4.11.3.7 char* pHTS_getLabelFromFile (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get one label string from a label file.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a pointer to the new label string retrieved from the label file.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.11.3.8 int pHTS_getNumberOfLabels (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the number of labels from a label file.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

an integer for the number of labels from a label file.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.9 int pHTS_getNumberOfSamples (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the number of the generated samples.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

an integer for the number of the generated samples.

Attention

You must generate the speech samples of the last parsed label in the pHTS environment using pHTS_updateSamples before calling this function.

4.11.3.10 HTS_PStreamSet* pHTS_getPDFs (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the set of PDF streams.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a HTS_Label pointer to the PDF stream set.

Attention

You must generate the PDFs of the last parsed label in the pHTS environment using pHTS_updatePDFs before calling this function.

4.11.3.11 double pHTS_getPitch (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the pitch value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the pitch value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.12 void pHTS_getSamples (HTS_Engine * *engine*, pHTS * *p*, short * *buffer*, int *nOfSamples*)

A streaming environment method to get the generated speech samples.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.
buffer	short pointer to the buffer to be filled with the generated samples.
nOfSamples	an integer for the number of samples to be copied in in the buffer.

Attention

You must generate the speech samples of the last parsed label in the pHTS environment using pHTS_updateSamples before calling this function.

4.11.3.13 double pHTS_getSpeed (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the speech speed rate.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the speech speed rate.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.14 double pHTS_getVolume (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to get the volume value.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.

Returns

a double for the volume value.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.15 void pHTS_initialize (int *num_arg*, char ** *arguments*, HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to initialize all environment variables.

Parameters

num_arg	an integer for the number of arguments (argc).
arguments	a character pointer to all arguments (argv).
engine	a HTS_Engine pointer to be initialized.
p	a pHTS pointer to be initialized.

Attention

To start the pHTS environment, first you must call this function.

4.11.3.16 int pHTS_parseLabel (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to parse the last inserted label.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Returns

an integer, 0 if the label ring buffer is full, and -1 is there is still space.

Attention

You must push a label in the pHTS environment using pHTS_pushLabel before calling this function.

4.11.3.17 void pHTS_popSamples (HTS_Engine * *engine*, pHTS * *p*, float * *buffer*, int *nOfSamples*)

A streaming environment method to access and pop the generated speech samples from ring buffer.

Parameters

engine	a HTS_Engine pointer to be accessed.
p	a pHTS pointer to be accessed.
buffer	float pointer to the buffer to be filled with the generated samples.
nOfSamples	an integer for the number of samples to be poped out of the ring buffer.

Attention

Must be called from audio callback

4.11.3.18 void pHTS_pushLabel (HTS_Engine * *engine*, pHTS * *p*, char * *label*)

A streaming environment method to set a new label into the labels set in order to be prosessed.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
label	character pointer to a phonetic label.

Attention

You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.19 void pHTS_refresh (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to flush the internal memory of the engine occupied by the pHTS environment.

Parameters

engine	a HTS_Engine pointer to be accessed and refreshed.
p	a pHTS pointer to be accessed.

Attention

You must initialize the pHTS environment and the HTS engine using pHTS_initialize before calling this function.

4.11.3.20 void pHTS_setAlpha (HTS_Engine * *engine*, pHTS * *p*, double *alpha*)

A streaming environment method to set different alpha value, the all-pass constant.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
alpha	a double for the new alpha, all-pass constant value.

Attention

Values less than zero or greater then one are ignored. Default value is 0.42.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.21 void pHTS_setDuration (HTS_Engine * *engine*, pHTS * *p*, double *duration*, int *mode*)

A streaming environment method to set different duration value in the generated speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
duration	a double for the new duration value.
mode	an integer to determine the action applied on the duration trajectory, if mode=0, the existing trajectory is overwritten with the value of duration, if mode=1, the existing trajectory is shifted up if duration>0 or shifted down if duration<0, if mode=2, the existing trajectory is scaled.

Attention

Values less than zero are ignored if mode=0 or mode=2.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.22 void pHTS_setFilter (HTS_Engine * *engine*, pHTS * *p*, HTS_GStreamSet * *gss*)

A streaming environment method to set a new set of generated parameter stream.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
gss	a HTS_GStreamSet pointer to the new set of generated parameter stream.

Attention

You must initialize the pHTS environment using pHTS_initialize and also create a valid set of generated parameters before calling this function.

4.11.3.23 void pHTS_setLabels (HTS_Engine * *engine*, pHTS * *p*, HTS_Label * *label*)

A streaming environment method to set a new set of labels.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
label	a HTS_Label pointer to the new label set.

Attention

You must initialize the pHTS environment using pHTS_initialize and also create a valid set of labels before calling this function.

4.11.3.24 void pHTS_setPDFs (HTS_Engine * *engine*, pHTS * *p*, HTS_PStreamSet * *pss*)

A streaming environment method to set a new set of PDF streams.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
pss	a HTS_PStreamSet pointer to the new PDF streams set.

Attention

You must initialize the pHTS environment using pHTS_initialize and also create a valid set of PDFs before calling this function.

4.11.3.25 void pHTS_setPitch (HTS_Engine * *engine*, pHTS * *p*, double *pitch*, int *mode*)

A streaming environment method to set different pitch value in the generated speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
pitch	a double for the new pitch value.
mode	an integer to determine the action applied on the pitch trajectory, if mode=0, the existing trajectory is overwritten with the value of pitch, if mode=1, the existing trajectory is shifted up if pitch>0 or shifted down if pitch<0.

Attention

the engine is using log f0.
values less than zero are ignored if mode=0.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.26 void pHTS_setSpeed (HTS_Engine * *engine*, pHTS * *p*, double *speechSpeed*)

A streaming environment method to set different speech speed rate.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
speechSpeed	a double for the new speech speed rate.

Attention

Values less than zero are ignored. Default value is 1.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.27 void pHTS_setVolume (HTS_Engine * *engine*, pHTS * *p*, double *volume*)

A streaming environment method to set different volume.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.
volume	a double for the new volume.

Attention

Values less than zero are ignored. Default value is 1.
You must initialize the pHTS environment using pHTS_initialize before calling this function.

4.11.3.28 void pHTS_updateEngine (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the speech samples from PDFs.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must parse the last pushed label in the pHTS environment using pHTS_-parseLabel before calling this function.

4.11.3.29 void pHTS_updateFilter (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the speech parameters from the PDFs.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must generate the PDFs of the last parsed label in the pHTS environment using pHTS_updatePDFs before calling this function.

4.11.3.30 void pHTS_updatePDFs (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the PDFs.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must parse the last pushed label in the pHTS environment using pHTS_-parseLabel before calling this function.

4.11.3.31 void pHTS_updateSamples (HTS_Engine * *engine*, pHTS * *p*)

A streaming environment method to generate the speech samples from the speech parameters.

Parameters

engine	a HTS_Engine pointer to be updated.
p	a pHTS pointer to be updated.

Attention

You must generate the speech parameters from the PDFs of the last parsed label in the pHTS environment using pHTS_updateFilter before calling this function.

Index