8.15 libjulius/include/julius/callback.h File Reference

Definitions for callback API.

Defines

- #define MAX_CALLBACK_HOOK 10

  Maximum number of callbacks that can be registered for each ID.

Enumerations

- enum {

  CALLBACK_POLL, CALLBACK_EVENT_PROCESS_ONLINE, CALLBACK_EVENT_PROCESS_OFFLINE, CALLBACK_EVENT_STREAM_BEGIN,

  CALLBACK_EVENT_STREAM_END, CALLBACK_EVENT_SPEECH_READY, CALLBACK_EVENT_SPEECH_START, CALLBACK_EVENT_SPEECH_STOP,

  CALLBACK_EVENT_RECOGNITION_BEGIN, CALLBACK_EVENT_RECOGNITION_END, CALLBACK_EVENT_SEGMENT_BEGIN, CALLBACK_EVENT_SEGMENT_END,

  CALLBACK_EVENT_PASS1_BEGIN, CALLBACK_EVENT_PASS1_FRAME, CALLBACK_EVENT_PASS1_END, CALLBACK_RESULT_PASS1_INTERIM,

  CALLBACK_RESULT_PASS1, CALLBACK_RESULT_PASS1_GRAPH, CALLBACK_STATUS_PARAM, CALLBACK_EVENT_PASS2_BEGIN,

  CALLBACK_EVENT_PASS2_END, CALLBACK_RESULT, CALLBACK_RESULT_GMM, CALLBACK_RESULT_GRAPH,

  CALLBACK_RESULT_CONFNET, CALLBACK_ADIN_CAPTURED, CALLBACK_ADIN_TRIGGERED, CALLBACK_EVENT_PAUSE,

  CALLBACK_EVENT_RESUME, CALLBACK_PAUSE_FUNCTION, **CALLBACK_DEBUG_PASS2_POP**, **CALLBACK_DEBUG_PASS2_PUSH**,

  CALLBACK_RESULT_PASS1_DETERMINED, **SIZEOF_CALLBACK_ID** }

  Callback IDs.

8.15.1 Detailed Description

Definitions for callback API.

Author:

Akinobu Lee

Date:

Mon Nov 5 18:30:04 2007

$Revision:$

Definition in file callback.h.

8.15.2 Enumeration Type Documentation

8.15.2.1 anonymous enum

Callback IDs.

Enumerator:

CALLBACK_POLL Callback to be called periodically while recognition.

CALLBACK_EVENT_PROCESS_ONLINE Event callback to be called when the engine becomes active and start running.

(ex. resume by j_request_resume())

CALLBACK_EVENT_PROCESS_OFFLINE Event callback to be called when the engine becomes inactive and stop running.

(ex. pause or terminate by user request)

CALLBACK_EVENT_STREAM_BEGIN (not implemented yet)

CALLBACK_EVENT_STREAM_END (not implemented yet)

CALLBACK_EVENT_SPEECH_READY Event callback to be called when engine is ready for recognition and start listening to the audio input.

CALLBACK_EVENT_SPEECH_START Event callback to be called when input speech processing starts.

This will be called at speech up-trigger detection by level and zerocross. When the detection is disabled (i.e. file input), This will be called immediately after opening the file.

CALLBACK_EVENT_SPEECH_STOP Event callback to be called when input speech ends.

This will be called at speech down-trigger detection by level and zerocross. When the detection is disabled (i.e. file input), this will be called just after the whole input has been read.

CALLBACK_EVENT_RECOGNITION_BEGIN Event callback to be called when a valid input segment has been found and speech recognition process starts.

This can be used to know the actual start timing of recognition process. On short-pause segmentation mode and decoder-based VAD mode, this will be called only once at a triggered long input.

See also:

CALLBACK_EVENT_SEGMENT_BEGIN.

CALLBACK_EVENT_RECOGNITION_END Event callback to be called when a valid input segment has ended up, speech recognition process ends and return to wait status for another input to come.

On short-pause segmentation mode and decoder-based VAD mode, this will be called only once after a triggered long input.

See also:

CALLBACK_EVENT_SEGMENT_END.

CALLBACK_EVENT_SEGMENT_BEGIN On short-pause segmentation and decoder-based VAD mode, this callback will be called at the beginning of each segment, segmented by short pauses.

CALLBACK_EVENT_SEGMENT_END On short-pause segmentation and decoder-based VAD mode, this callback will be called at the end of each segment, segmented by short pauses.

CALLBACK_EVENT_PASS1_BEGIN Event callback to be called when the 1st pass of recognition process starts for the input.

CALLBACK_EVENT_PASS1_FRAME Event callback to be called periodically at every input frame.

This can be used to get progress status of the first pass at each frame.

CALLBACK_EVENT_PASS1_END Event callback to be called when the 1st pass of recognition process ends for the input and proceed to 2nd pass.

CALLBACK_RESULT_PASS1_INTERIM Result callback to be called periodically at the 1st pass of recognition process, to get progressive output.

CALLBACK_RESULT_PASS1 Result callback to be called just at the end of 1st pass, to provide recognition status and result of the 1st pass.

CALLBACK_RESULT_PASS1_GRAPH When compiled with "–enable-word-graph", this callback will be called at the end of 1st pass to provide word graph generated at the 1st pass.

CALLBACK_STATUS_PARAM Status callback to be called after the 1st pass to provide information about input (length etc.

)

CALLBACK_EVENT_PASS2_BEGIN Event callback to be called when the 2nd pass of recognition process starts.

CALLBACK_EVENT_PASS2_END Event callback to be called when the 2nd pass of recognition process ends.

CALLBACK_RESULT Result callback to provide final recognition result and status.

CALLBACK_RESULT_GMM Result callback to provide result of GMM computation, if GMM is used.

CALLBACK_RESULT_GRAPH Result callback to provide the whole word lattice generated at the 2nd pass.

Use with "-lattice" option.

CALLBACK_RESULT_CONFNET Result callback to provide the whole confusion network generated at the 2nd pass.

Use with "-confnet" option.

CALLBACK_ADIN_CAPTURED A/D-in plugin callback to access to the captured input.

This will be called at every time a small audio fragment has been read into Julius. This callback will be processed first in Julius, and after that Julius will process the content for recognition. This callback can be used to monitor or modify the raw audio input in user-side application.

CALLBACK_ADIN_TRIGGERED A/D-in plugin callback to access to the triggered input.

This will be called for input segments triggered by level and zerocross. After processing this callback, Julius will process the content for recognition. This callback can be used to monitor or modify the triggered audio input in user-side application.

CALLBACK_EVENT_PAUSE Event callback to be called when the engine becomes paused.

CALLBACK_EVENT_RESUME Event callback to be called when the engine becomes resumed.

CALLBACK_PAUSE_FUNCTION Plugin callback that will be called inside Julius when the engine becomes paused.

When Julius engine is required to stop by user application, Julius interrupu the recognition and start calling the functions registered here. After all the functions are executed, Julius will resume to the recognition loop. So if you want to use the pause / resume facility of Julius, You should also set callback function to this to stop and do something, else Julius returns immediately.

Definition at line 32 of file callback.h.