**VARender Instructions**

# 1 Overview

The library is an encapsulated audio/video decoding and playing library under a windows system. The main functions are as follows:

1. The decoding of audio and video, support multiple algorithms, and support decoding naked data callbacks.
2. The playing of audio and video, support the synchronization of audio and video.

# 2 Main functions and using steps

## 2.1 Global initialization and anti-initialization of library

1) VARender\_Startup, call the function first after the program starts, other functions of the library cannot be called before the call.

2) VARender\_Cleanup, call the function finally at the end of the program, other functions of the library cannot be called after the call.

## 2.2 Creation and destruction of audio/video decoding and playing objects

1) VARender\_Open, firstly use the function to create an object handle (VARENDER\_HANDLE) before decoding and playing.

2) VARender\_Close, use the function to close the object at the end of the decoding and playing.

## 2.3 Specified windows and areas for video playing

1) VARender\_SetWindow, use the function to specify a window to play before plugging into video data, otherwise the video cannot be displayed.

2) VARender\_SetDestRect, specify the place in a window for video playing. Default full-window play on the specified window handle if the function is not called, and the playing area automatically follows the change of the window size.

3) The handle and area of the playing window can be changed at any time during the playing process and take effect immediately.

## 2.4 Plugging in audio and video data

1) VARender\_PumpVideoFrame, using the video frame data received by the functions of other libraries in our SDK to plug into the library with the function, and the video can be played on the specified window. Note: the video frame is not naked bit stream, it has our private frame header in front.

2) VARender\_PumpAudioFrame, using the audio frame data received by the functions of other libraries in our SDK to plug into the library with the function, and the sound can be played in the default audio player device, such as earphone or loudspeaker. Note: the audio frame is not naked bit stream, it has our private frame header in front.

# 3 Advanced function instruction

Note: without special instructions, all of the following advanced functions can be called during the playing process and take effect immediately.

## 3.1 How to enable hardware decoding

1) VARender\_EnableHA, the video decoding of the library uses soft decoding by default, if want to use hardware decoding, the function can be called to enable or close. Enable the hardware decoding will use the hardware decoding function of video card, and the video may not be decoded or abnormal display of the decoded video for the hardware mismatch.

## 3.2 how to save the last frame image when repainting window

1) VARender\_Refresh, call the function in the window repainting functions (such as ON\_PAINT function in VC).

2) VARender\_EnableRefresh，sometimes we do not want to show the last frame image when repainting the window, the function can be called to disable it.

## 3.3 How to obtain the decoded audio and video data

1) VARender\_SetVideoCallBack, set up a callback function via the function to get the decoded video YUV or RGB data. The detailed format of the video can be specified by cFormat parameter.

2) VARender\_SetAudioCallBack, set up a callback function via the function to get the decoded audio PCM data.

## 3.4 How to overlay characters and other information on the displayed video

1) VARender\_SetDisplayCallBack, set up a callback function via the function to get a window DC handle, call the API of Windows system via the handle to overlay text or draw graphics on the DC.

## 3.5 How to display video on other monitor for multi-monitor

1) VARender\_SetMonitor, when multiple monitors form a large desktop, if DDraw is enabled, the function needs to be called to specify monitor for the video displayed on other monitors, otherwise the video cannot be displayed. If DDraw is disabled, there is no need to call the function.

## 3.6 How to specify only one cannel to output sound between left and right sound channels

1) VARender\_SetAudioLRChannel, set the left or right sound channel to output sound with the function, two sound channels to output simultaneously by default if not specified.

## 3.7 How to set video to display as original proportion

1) VARender\_SetFixAspectRatio, call the function to specify video is shown as original proportion or not. The default is to stretch according to the size of playing area.

## 3.8 How to realize electronic amplification function

1) VARender\_SetSrcRect, call the function to specify displaying a certain area in the original video to realize the digital amplification function.

## 3.9 How to enable or close DDraw function

1. VARender\_SetDDraw, call the function to close or enable DDraw.
2. VARender\_IsDDraw, call the function to judge if DDraw is currently enabled.

## 3.10 How to realize pause and resumption of video and audio during playing

1. The pause and resume playing of video and audio can be realized via calling the following 4 functions:
   1. VARender\_StartVideo
   2. VARender\_StopVideo
   3. VARender\_StartAudio
   4. VARender\_StopAudio
2. Because VARender library is mainly used to play real-time video, decoding will continue during pause period, and the period of audio and video would be skipped after resuming to play. In this sense, it appears that there is no practical use for these functions.
3. For replay video file, pause and resumption still need the upper level to control plugging into data or not.

## 3.11 How to capture the displaying video and save it to picture file during playing

1) VARender\_Snapshot, call the function to capture pictures, the suffix must be BMP.

## 3.12 How to set the audio and video to be played synchronously or not

1) VARender\_SetSync, call the function to set the audio and video to be played synchronously or not.

2) VARender\_IsSync, call the function to determine if the current audio and video must be played synchronously.

## 3.13 How to set the play mode to play the video file

1) VARender\_SetPlayMode, call the function to set play mode, default mode is trial, play video needs to set as playback mode.

2) The difference between two modes is as follows:

a) Real-time mode

* 1. Dynamically adjust play speed according to buffered data size.
  2. The accurate synchronization of audio and video is determined by VARender\_SetSync function, the default is not accurate.

b) Playback mode

* 1. Play strictly follow the timestamp and original speed, and audio and video can realize accurate synchronization.
  2. VARender\_SetSpeed can be used to change the play speed by integer multiples.
  3. There are little internal buffered frames for the mode, the upper level needs to plug into data in time.

3) VARender\_GetPlayMode, call the function to obtain the current play mode.

4) VARender\_GetSpeed, call the function to obtain the current play speed.

# 4 Detailed instructions for the functions and parameters

Please refer to the instructions of file <VARender.h>.