



Client Audio Encoding and Decoding Library

## API Reference

|                    |            |
|--------------------|------------|
| <b>Issue</b>       | 05         |
| <b>Date</b>        | 2008-11-30 |
| <b>Part Number</b> | N/A        |

HiSilicon Technologies CO., LIMITED. provides customers with comprehensive technical support and service. Please feel free to contact our local office or company headquarters.

## HiSilicon Technologies CO., LIMITED.

Address: Manufacture Center of Huawei Electric, Huawei Base,  
Bantian, Longgang District, Shenzhen, 518129, People's Republic of China

Website: <http://www.hisilicon.com>

Tel: +86-755-28788858


Fax: +86-755-28357515

Email: [support@hisilicon.com](mailto:support@hisilicon.com)

**Copyright © HiSilicon Technologies CO., LIMITED. 2007-2008. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon Technologies CO., LIMITED.

### Trademarks and Permissions

 **HISILICON**, and other HiSilicon icons are trademarks of HiSilicon Technologies CO., LIMITED.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

### Notice

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied.



# Contents

|   |            |
|---|------------|
| <b>About This Document.....</b>                       | <b>1</b>   |
| <b>1 Overview.....</b>                                | <b>1-1</b> |
| 1.1 Scope .....                                       | 1-1        |
| 1.2 API Format .....                                  | 1-1        |
| 1.3 API List .....                                    | 1-1        |
| 1.4 API Reference Fields.....                         | 1-2        |
| 1.5 Data Structure Reference Fields .....             | 1-2        |
| 1.6 Structure of the Hisilicon Voice Frame.....       | 1-3        |
| <b>2 API References .....</b>                         | <b>2-1</b> |
| 2.1 Initializing the Hisilicon Voice Encoder .....    | 2-1        |
| 2.2 Initializing the Hisilicon Voice Decoder.....     | 2-2        |
| 2.3 Hisilicon Voice Codec .....                       | 2-4        |
| 2.4 Initializing the Hisilicon Voice Transcoder ..... | 2-7        |
| 2.5 Hisilicon Voice Transcoder.....                   | 2-8        |
| <b>3 Other Information.....</b>                       | <b>3-1</b> |
| 3.1 Data Types .....                                  | 3-1        |
| 3.2 Error Codes .....                                 | 3-4        |
| <b>A Acronyms and Abbreviations.....</b>              | <b>A-1</b> |



---

## Tables

---

|   |     |
|---|-----|
| <b>Table 1-1</b> API list .....                               | 1-1 |
| <b>Table 1-2</b> API reference fields .....                   | 1-2 |
| <b>Table 1-3</b> Data structure reference fields .....        | 1-2 |
| <b>Table 1-4</b> Structure of the Hisilicon voice frame ..... | 1-3 |
| <b>Table 3-1</b> Error codes.....                             | 3-4 |



# About This Document

## Purpose

This document describes the document contents, related product versions, intended audience, conventions and update history.

## Related Versions

The following table lists the product versions related to this document.

| Product Name                                 | Version |
|--|---------|
| Hi3510 Communications Media Processor        | V100    |
| Hi3511 H.264 Encoding and Decoding Processor | V100    |
| Hi3512 H.264 Encoding and Decoding Processor | V100    |

## Intended Audience

This document is intended for the programmers who have a good command of the C language.

## Organization

This document is organized as follows:

| Chapter             | Description   |
|---------------------|---|
| 1 Overview          | Describes the API reference fields of the client audio encoding and decoding library. |
| 2 API References    | Describes the APIs in the document in detail.   |
| 3 Other Information | Provides other information such as data structures and error codes.                   |








| Chapter                               | Description  |
|---------------------------------------|--|
| Appendix A Acronyms and Abbreviations | Lists the abbreviations and acronyms and gives their full spellings. |

## Conventions

### Symbol Conventions

The following symbols may be found in this document. They are defined as follows.

| Symbol   | Description   |
|--|---|
|  <b>DANGER</b>    | Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.   |
|  <b>WARNING</b>  | Indicates a hazard with a medium or low level of risk that, if not avoided, could result in minor or moderate injury.   |
|  <b>CAUTION</b> | Indicates a potentially hazardous situation that, if not avoided, could cause equipment or component damage, data loss, and performance degradation, or unexpected results. |
|  <b>TIP</b>     | Indicates a tip that may help you solve a problem or save time.   |
|  <b>NOTE</b>    | Provides additional information to emphasize or supplement important points of the main text.   |

### General Conventions

| Convention      | Description  |
|-----------------|--|
| Times New Roman | Normal paragraphs are in Times New Roman.  |
| <b>Boldface</b> | Names of files, directories, folders, and users are in <b>boldface</b> . For example, log in as user <b>root</b> . |
| <i>Italic</i>   | Book titles are in <i>italics</i> .  |
| Courier New     | Terminal display is in Courier New.  |



## Update History

Updates between document versions are cumulative. Therefore, the latest document version contains all updates made to previous versions.

| Date       | Revision | Description   |
|------------|----------|---|
| 2008-11-27 | 05       | Information about the Hi3511 and Hi3512 is added.   |
| 2007-12-18 | 04       | <ul style="list-style-type: none"><li>• HI_VOICE_TransCodeReset and HI_VOICE_TransCodeFrame are added to section 1.3 "API List."</li><li>• The note of HI_VOICE_EncReset is modified.</li><li>• The note of HI_VOICE_DecReset is modified.</li><li>• Section 2.4 "Initializing the Converter" is added.</li><li>• Section 2.5 "Converter" is added.</li><li>• Section 3.1.1 "Definitions of Constants" is modified. The codec type of the DVS1.3.1.0 version, converter type, and definitions of two error codes are returned.</li><li>• Section 3.1.5 "hiVOICE_TRANSCODE_STATE_S" is added.</li><li>• HI_ERR_VOICE_TRANS_TYPE and HI_ERR_VOICE_TRANS_DEVICE are added to section 3.2 "Error Codes" that is modified.</li></ul> |
| 2007-04-20 | 03       | <ul style="list-style-type: none"><li>• The note of HI_VOICE_EncodeFrame is modified.</li><li>• The codes in section 3.1.1 "Definitions of Constants" is modified.</li><li>• The codes in section 3.1.2 "hiVOICE_G711_STATE_S", 3.1.3 "hiVOICE_ADPCM_STATE_S", and 3.1.4 "hiVOICE_G726_STATE_S" are modified.</li><li>• Header files are modified.</li><li>• The note of HI_VOICE_EncodeFrame is modified.</li><li>• The note of HI_VOICE_DecodeFrame is modified.</li></ul>  |
| 2006-12-29 | 02       | <ul style="list-style-type: none"><li>• "API interface" has been changed to "API reference".</li><li>• The description of HI_VOICE_EncReset is modified.</li><li>• "typedef unsigned char HI_U8" before HI_VOICE_DecReset is deleted.</li><li>• The purpose, description of HI_VOICE_EncodeFrame is modified, and the note of HI_VOICE_EncodeFrame is added.</li><li>• The description, parameters of HI_VOICE_DecodeFrame is modified, and the note of HI_VOICE_DecodeFrame is added.</li><li>• "Data Type" and "Error Codes" has been combined into one Chapter, that is, Chapter 3.</li></ul>  |
| 2006-05-16 | 01       | Initial release   |



# 1 Overview

## 1.1 Scope

The client audio library supports the G.711, ADPCM, and G.726 codec. All the codecs are specified with only 8 kHz sampling rate. The G.711 Codec provides both  $\mu$  law and A law companding and uses the non-linear quantizing method to compress the sampling data from 128 kbit/s to 64 kbit/s. The G.711 is used in the integrated service digital network (ISDN) and most digital telephone backbones. The  $\mu$  law encoding is commonly used in North America and Japan. The A law is used throughout Europe and elsewhere. The adaptive differential pulse code modulation (ADPCM) coder compresses 128 kbit/s linear data to 40, 32, 24, and 16 kbit/s code using the ADPCM technique.

## 1.2 API Format

None.

## 1.3 API List

Table 1-1 lists the APIs in the document.

**Table 1-1** API list

| Function                             | Description  | Page                |
|--------------------------------------|--|---------------------|
| <a href="#">HI_VOICE_EncReset</a>    | Initialize the Hisilicon voice encoder.  | <a href="#">2-1</a> |
| <a href="#">HI_VOICE_DecReset</a>    | Initialize the Hisilicon voice decoder.  | <a href="#">2-2</a> |
| <a href="#">HI_VOICE_EncodeFrame</a> | Encode the audio input signal into a coded output packed in the Hisilicon voice frame. | <a href="#">2-4</a> |
| <a href="#">HI_VOICE_DecodeFrame</a> | Decode the Hisilicon voice frame.  | <a href="#">2-5</a> |





| Function                                | Description                                | Page                |
|---|--|---------------------|
| <a href="#">HI_VOICE_TransCodeReset</a> | Initialize the Hisilicon voice transcoder. | <a href="#">2-7</a> |
| <a href="#">HI_VOICE_TransCodeFrame</a> | Transcode the Hisilicon voice frame.       | <a href="#">2-8</a> |

## 1.4 API Reference Fields

This document describes APIs in the following ten reference fields. See [Table 1-2](#).

**Table 1-2** API reference fields

| Parameter Field | Function  |
|-----------------|---|
| Purpose         | Describes the major function of an API.   |
| Syntax          | Lists the syntax of an API.   |
| Description     | Describes the working process of an API.  |
| Parameter       | Lists the parameters and the related information of an API.   |
| Return Value    | Lists the return values and the related information of an API.  |
| Error Code      | Lists the error codes and the related information of an API.  |
| Request         | Lists the header files contained in an API and the library files linked to an API when the API is compiled. |
| Note            | Describes the matters that you need pay attention to when using an API.                                     |
| Example         | Provides the example of using an API.   |
| See Also        | Describes other information about an API.   |

## 1.5 Data Structure Reference Fields

This document describes data structures in the following three reference fields. See [Table 1-3](#).

**Table 1-3** Data structure reference fields

| Parameter Field | Function   |
|-----------------|--|
| Description     | Describes the functions implemented by a data structure. |
| Definition      | Defines a data structure.                                |



| Parameter Field | Function  |
|-----------------|---|
| Note            | Describes the matters that you need pay attention to when using a data structure. |

## 1.6 Structure of the Hisilicon Voice Frame

[Table 1-4](#) shows the structure of the Hisilicon voice frame.

**Table 1-4** Structure of the Hisilicon voice frame

| Parameter Position (Unit: HI_S16) | Bits   | Description  |
|-----------------------------------|--------|--|
| 0                                 | [15:8] | Flag of a frame type.<br>01: Hisilicon voice frame.<br>Others: reserved. |
|                                   | [7:0]  | Reserved.  |
| 1                                 | [15:8] | Frame counter: 0–255.  |
|                                   | [7:0]  | Length of the speech data, in the unit of HI_S16.                        |
| 2                                 | [15:0] | Speech data.   |
| 3                                 | [15:0] | Speech data.   |
| ...                               | [15:0] | Speech data.   |
| 2+n-2                             | [15:0] | Speech data.   |
| 2+n-1                             | [15:0] | Speech data.   |

Note: For the value range of n, see section [2.3 "Hisilicon Voice Codec."](#)



# 2 API References

## 2.1 Initializing the Hisilicon Voice Encoder

### HI\_VOICE\_EncReset

[Purpose]

Initialize the Hisilicon voice encoder.

[Syntax]

```
#include "hi_voice_api.h"

HI_VOICE_API HI_RESULT HI_VOICE_EncReset (HI_VOID *pEncState,
HI_S16 nCoder);
```

[Description]

The API is used to initialize the state control data structure for the encoder and must be called first when the channel is first created.

[Parameter]

| Parameter Name | Description   | Input/Output | Global/Local |
|----------------|---|--------------|--------------|
| pEncState      | Specified encoder. For details, see section 3.1 <a href="#">"hiVOICE_G711_STATE_S"</a> , 3.1 <a href="#">"hiVOICE_ADPCM_STATE_S"</a> , or 3.1 <a href="#">"hiVOICE_G726_STATE_S."</a> | Input/Output | Global       |
| nCoder         | Specified encoder type. For details, see section 3.1 <a href="#">"Definitions of Constants."</a>  | Input        | Local        |

[Return Value]

| Return Value | Description |
|--------------|-------------|
| 0            | Success.    |



| Return Value | Description                                 |
|--------------|---|
| Non-zero     | Failure. The return value is an error code. |

## [Error Code]

| Error Code                                  | Description                            |
|---|--|
| <a href="#">HI_ERR_VOICE_ENC_TYPE</a>       | The encoder type is invalid.           |
| <a href="#">HI_ERR_VOICE_INVALID_DEVICE</a> | The pointer to the encoder is invalid. |

## [Request]

Header file: `/include/hi_voice_api.h`.

## [Note]

The G711\_ORG\_A, G711\_ORG\_U, ADPCM\_ORG\_DVI4, and G726\_ORG\_16KBPS encoder types are defined only for the Hi3510\_VSSDK\_V1.3.1.0. It is not recommended to use the four encoder types in the later versions.

## [Example]

```
HI_VOID VoiceDemo()  
{  
    HI_RESULT Ret;  
    hiVOICE_G711_STATE_S g_sG711EncState;  
    Ret =HI_VOICE_EncReset (&g_sG711EncState, G711_A);  
    if (Ret != 0)  
        MyErrorExit("HISI_VOICE_enc_reset ");  
    else  
        printf("HISI_VOICE_enc_reset SUCCESS.\n");  
}
```

## [See Also]

[HI\\_VOICE\\_EncodeFrame](#).

## 2.2 Initializing the Hisilicon Voice Decoder

### HI\_VOICE\_DecReset

## [Purpose]

Initialize the Hisilicon voice decoder.

## [Syntax]

```
#include "hi_voice_api.h"
```



```
HI_VOICE_API HI_RESULT HI_VOICE_DecReset (HI_VOID *pDecState,  
HI_S16 nCoder);
```

[Description]

The API is used to initialize the state control data structure for the decoder and must be called first when the channel is first created.

[Parameter]

| Parameter Name | Description  | Input/Output | Global/Local |
|----------------|--|--------------|--------------|
| pDecState      | Specified decoder. For details, see section 3.1 <a href="#">"hiVOICE_G711_STATE_S"</a> , 3.1 <a href="#">"hiVOICE_ADPCM_STATE_S"</a> , or 3.1 <a href="#">"hiVOICE_G726_STATE_S"</a> . | Input/Output | Global       |
| nCoder         | Specified decoder type. For details, see section 3.1 <a href="#">"Definitions of Constants."</a>   | Input        | Local        |

[Return Value]

| Return Value | Description                                 |
|--------------|---|
| 0            | Success.                                    |
| Non-zero     | Failure. The return value is an error code. |

[Error Code]

| Error Code                                  | Description                            |
|---|--|
| <a href="#">HI_ERR_VOICE_DEC_TYPE</a>       | The decoder type is invalid.           |
| <a href="#">HI_ERR_VOICE_INVALID_DEVICE</a> | The pointer to the decoder is invalid. |

[Request]

Header file: `/include/hi_voice_api.h`.

[Note]

The G711\_ORG\_A, G711\_ORG\_U, ADPCM\_ORG\_DVI4, and G726\_ORG\_16KBPS decoder types are defined only for the Hi3510\_VSSDK\_V1.3.1.0. It is not recommended to use the four decoder types in the later versions.

[Example]

```
HI_VOID VoiceDemo()  
{  
    HI_RESULT Ret;
```



```
hiVOICE_G711_STATE_S g_sG711DecState;  
Ret=HI_VOICE_DecReset (&g_sG711DecState, G711_A);  
if (Ret != 0)  
MyErrorExit("HISI_VOICE_dec_reset");  
else  
printf("HISI_VOICE_dec_reset SUCCESS.\n");  
}
```

[See Also]

[HI\\_VOICE\\_DecodeFrame](#).

## 2.3 Hisilicon Voice Codec

HI\_VOICE\_EncodeFrame() is used to encode the raw audio data and then output the Hisilicon voice frames. HI\_VOICE\_DecodeFrame() is used to decode the Hisilicon voice frames and output the decoded audio data.

### HI\_VOICE\_EncodeFrame

[Purpose]

Encode the audio input signal into a coded output packed in the Hisilicon voice frame.

[Syntax]

```
#include "hi_voice_api.h"  
HI_VOICE_API HI_RESULT HI_VOICE_EncodeFrame(HI_VOID *pEncState,  
HI_S16 *pInputBuf,  
HI_S16 *pOutputBuf,  
HI_S16 len);
```

[Description]

The API is used to encode the 16-bit audio input signal into a coded output pointed to by pOutputBuf, and the output is packed in the Hisilicon voice frame. Note that the maximum frame length is HI\_VOICE\_MAX\_FRAME\_SIZE. Generally, the frame length is a multiple of 80, namely, 80, 160, 240, 320, or 480. But for the ADPCM\_IMA codec, an extra sampling point that functions as the reference level is required, so that the frame length is 81, 161, 241, 321, or 481.

[Parameter]

| Parameter Name | Description                                    | Input/Output | Global/Local |
|----------------|--|--------------|--------------|
| pEncState      | Specified encoder.                             | Input/Output | Global       |
| pInputBuf      | Pointer to the input audio data to be encoded. | Input        | Local        |
| pOutputBuf     | Pointer to the coded data.                     | Output       | Local        |



| Parameter Name | Description | Input/Output | Global/Local |
|----------------|-------------|--------------|--------------|
| len            | Frame size. | Input        | Local        |

[Return Value]

| Return Value | Description                                 |
|--------------|---|
| 0            | Success.                                    |
| Non-zero     | Failure. The return value is an error code. |

[Error Code]

| Error Code                                  | Description                              |
|---|--|
| <a href="#">HI_ERR_VOICE_ENC_TYPE</a>       | The encoder type is invalid.             |
| <a href="#">HI_ERR_VOICE_INVALID_DEVICE</a> | The pointer to the encoder is invalid.   |
| <a href="#">HI_ERR_VOICE_INVALID_INBUF</a>  | The input buffer is invalid.             |
| <a href="#">HI_ERR_VOICE_INVALID_OUTBUF</a> | The output buffer is invalid.            |
| <a href="#">HI_ERR_VOICE_ENC_FRAME_SIZE</a> | The length of the data frame is invalid. |

[Request]

Header file: `/include/hi_voice_api.h`.

[Note]

Ensure that the encoder is initialized before an encoding process.

[Example]

```
HI_VOID VoiceDemo()  
{  
    Ret=HI_VOICE_EncodeFrame(&g_sG711EncState,in_pcm_buf,ou_enc_buf,MAX_FRAME  
_SIZE);  
}
```

[See Also]

- [HI\\_VOICE\\_EncReset](#).
- [Structure of the Hisilicon Voice Frame](#).

## HI\_VOICE\_DecodeFrame

[Purpose]



Decode the Hisilicon voice frame.

[Syntax]

```
#include "hi_voice_api.h"

HI_VOICE_API HI_RESULT HI_VOICE_DecodeFrame (HI_VOID *pDecState,
HI_S16 * pInputBuf,
HI_S16 * pOutputBuf,
HI_S16 * pLen);
```

[Description]

The API is used to transcode the coded input signal packed in the Hisilicon voice frame into a decoded output, and the output is the 16-bit linear data.

[Parameter]

| Parameter Name | Description                                    | Input/Output | Global/Local |
|----------------|--|--------------|--------------|
| pDecState      | Specified decoder.                             | Input/Output | Global       |
| pInputBuf      | Pointer to the input audio data to be decoded. | Input        | Local        |
| pOutputBuf     | Pointer to the coded data.                     | Output       | Local        |
| pLen           | Pointer to the length of the output data.      | Output       | Local        |

[Return Value]

| Return Value | Description                                 |
|--------------|---|
| 0            | Success.                                    |
| Non-zero     | Failure. The return value is an error code. |

[Error Code]

| Error Code                                  | Description                                |
|---|--|
| <a href="#">HI_ERR_VOICE_DEC_TYPE</a>       | The decoder type is invalid.               |
| <a href="#">HI_ERR_VOICE_INVALID_DEVICE</a> | The pointer to the decoder is invalid.     |
| <a href="#">HI_ERR_VOICE_INVALID_INBUF</a>  | The input buffer is invalid.               |
| <a href="#">HI_ERR_VOICE_INVALID_OUTBUF</a> | The output buffer is invalid.              |
| <a href="#">HI_ERR_VOICE_DEC_FRAMETYPE</a>  | The type of the audio frame is invalid.    |
| <a href="#">HI_ERR_VOICE_DEC_FRAME_SIZE</a> | The length of the decoded data is invalid. |





[Request]

Header file: /include/hi\_voice\_api.h.

[Note]

Ensure that the decoder is initialized and that the output buffer can accommodate a frame of decoded data before a decoding process. The compressed data is packed in the Hisilicon voice frame.

[Example]

```
HI_VOID VoiceDemo()  
{  
    Ret = HI_VOICE_DecodeFrame(&g_sg711DecState, in_unpacked, ou_dec_pcmbuf,  
    &len);  
}
```

[See Also]

- [HI\\_VOICE\\_DecReset.](#)
- [Structure of the Hisilicon Voice Frame.](#)

## 2.4 Initializing the Hisilicon Voice Transcoder

### HI\_VOICE\_TransCodeReset

[Purpose]

Initialize the Hisilicon voice transcoder.

[Syntax]

```
#include "hi_voice_api.h"  
HI_VOICE_API HI_RESULT HI_VOICE_TransCodeReset (HI_VOID *pTransState,  
HI_S16 nCoder);
```

[Description]

The API is used to initialize the Hisilicon voice transcoder.

[Parameter]

| Parameter Name | Description   | Input/Output | Global/Local |
|----------------|---|--------------|--------------|
| pTransState    | Specified transcoder.   | Input/Output | Global       |
| nCoder         | Type of the specified transcoder. The type of the specified transcoder is determined by the type of the Hi3510_VSSDK_V1.3.1.0 encoder. For details, see section 3.1 " <a href="#">Definitions of Constants</a> ." | Input        | Local        |



## [Return Value]

| Return Value | Description                                 |
|--------------|---|
| 0            | Success.                                    |
| Non-zero     | Failure. The return value is an error code. |

## [Error Code]

| Error Code                                | Description                               |
|---|---|
| <a href="#">HI_ERR_VOICE_TRANS_TYPE</a>   | The transcoder type is invalid.           |
| <a href="#">HI_ERR_VOICE_TRANS_DEVICE</a> | The pointer to the transcoder is invalid. |

## [Request]

Header file: `/include/hi_voice_api.h`.

## [Note]

The conversion process is available only for the Hi3510\_VSSDK\_V1.3.1.0 coded data.

## [Example]

```
HI_VOID VoiceDemo()  
{  
    HI_RESULT Ret;  
    hiVOICE_TRANSCODE_STATE_S g_TransState;  
    Ret= HI_VOICE_TransCodeReset (&g_TransState, ORG_G711A_TRANSF2_G711A);  
    if (Ret != 0)  
        MyErrorExit("HISI_VOICE_trans_reset");  
    else  
        printf("HISI_VOICE_trans_reset SUCCESS.\n");  
}
```

## [See Also]

[HI\\_VOICE\\_TransCodeFrame](#).

## 2.5 Hisilicon Voice Transcoder

The Hisilicon audio transcoder transcodes the G711A, G711U, ADPCM\_DVI4, or G726\_16KBPS coded data of the Hi3510\_VSSDK\_V1.3.1.0 version into that of a later version, and then outputs the transcoded data packed in the Hisilicon voice frame.



## HI\_VOICE\_TransCodeFrame

### [Purpose]

Transcode the Hisilicon voice frame.

### [Syntax]

```
#include "hi_voice_api.h"

HI_VOICE_API HI_RESULT HI_VOICE_TransCodeFrame (HI_VOID *pTransState,
HI_S16 *pInputBuf);
```

### [Description]

The API is used to transcode the Hisilicon voice frame and puts the transcoded data packed in the Hisilicon voice frame in the output buffer.

### [Parameter]

| Parameter Name | Description   | Input/Output | Global/Local |
|----------------|---|--------------|--------------|
| pTransState    | Specified transcoder.   | Input/Output | Global       |
| pInputBuf      | Pointer to the input audio data to be transcoded.<br>Pointer to the coded data. | Input/Output | Local        |

### [Return Value]

| Return Value | Description                                 |
|--------------|---|
| 0            | Success.                                    |
| Non-zero     | Failure. The return value is an error code. |

### [Error Code]

| Error Code                                 | Description                               |
|--|---|
| <a href="#">HI_ERR_VOICE_TRANS_TYPE</a>    | The transcoder type is invalid.           |
| <a href="#">HI_ERR_VOICE_TRANS_DEVICE</a>  | The pointer to the transcoder is invalid. |
| <a href="#">HI_ERR_VOICE_INVALID_INBUF</a> | The input buffer is invalid.              |

### [Request]

Header file: /include/hi\_voice\_api.h.

### [Note]

- Ensure that the transcoder is initialized before a conversion process.



- Ensure that the input coded data is of the Hi3510\_VSSDK\_V1.3.1.0 version.

[Example]

```
HI_VOID VoiceDemo()  
{  
    Ret=HI_VOICE_TransCodeFrame (&g_TransState, in_enc_buf);  
}
```

[See Also]

- [HI\\_VOICE\\_TransCodeReset.](#)
- [Structure of the Hisilicon Voice Frame.](#)



# 3 Other Information

## 3.1 Data Types

### Definitions of Constants

```
/* The maximum frame length of the encoder */
#define HI_VOICE_MAX_FRAME_SIZE      481

/* Type of the Hisilicon codec*/
/* Real-time transport protocol (RTP) */
/* 64kbit/s G.711 A, see RFC3551.txt 4.5.14 PCMA */
#define G711_A                        0x01
/* 64kbit/s G.711 U, see RFC3551.txt 4.5.14 PCMU */
#define G711_U                        0x02
/* 32kbit/s ADPCM (DVI4) for RTP; see DVI4 differs in three respects from the
IMA ADPCM at RFC3551.txt 4.5.1 DVI4 */
#define ADPCM_DVI4                    0x03
/* 16kbit/s G.726, see RFC3551.txt 4.5.4 G726-16 */
#define G726_16KBPS                    0x04
/* 24kbit/s G.726, see RFC3551.txt 4.5.4 G726-24 */
#define G726_24KBPS                    0x05
/* 32kbit/s G.726, see RFC3551.txt 4.5.4 G726-32 */
#define G726_32KBPS                    0x06
/* 40kbit/s G.726, see RFC3551.txt 4.5.4 G726-40 */
#define G726_40KBPS                    0x07
/* IMA ADPCM wave type, 32kbit/s ADPCM (IMA) */
#define ADPCM_IMA                      0x23
/*
An example of the packing scheme for G726-32 codewords is as shown and bit
A3 is the least significant bit of the first codeword:
RTP G726-32:
0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
```



```
+-+-+---+---+---+---+---+---+---+---+---+
|B B B B|A A A A|D D D D|C C C C| ...
|0 1 2 3|0 1 2 3|0 1 2 3|0 1 2 3|
+-+-+---+---+---+---+---+---+---+---+---+

MEDIA G726-32:
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+---+---+---+---+---+---+---+---+---+
|A A A A|B B B B|C C C C|D D D D| ...
|3 2 1 0|3 2 1 0|3 2 1 0|3 2 1 0|
+-+-+---+---+---+---+---+---+---+---+---+
*/

#define MEDIA_G726_16KBPS      0x24 /* G726 16kbit/s for ASF */
#define MEDIA_G726_24KBPS      0x25 /* G726 24kbit/s for ASF */
#define MEDIA_G726_32KBPS      0x26 /* G726 32kbit/s for ASF */
#define MEDIA_G726_40KBPS      0x27 /* G726 40kbit/s for ASF */

/* Type of the codec of the Hi3510_VSSDK_V1.3.1.0 version */
#define G711_ORG_A              0x41 /* 64kbit/s G.711 A */
#define G711_ORG_U              0x42 /* 64kbit/s G.711 U */
#define ADPCM_ORG_DVI4          0x43 /* 32kbit/s ADPCM */
#define G726_ORG_16KBPS         0x44 /* 16kbit/s G.726 */

/* Type of the transcoder */
/* transform Hi3510_VSSDK_V1.3.1.0 64kbit/s G.711 A to current G711_A*/
#define ORG_G711A_TRANSF2_G711A      0x61

/* transform Hi3510_VSSDK_V1.3.1.0 64kbit/s G.711 U to current G711_U*/
#define ORG_G711U_TRANSF2_G711U      0x62

/* transform Hi3510_VSSDK_V1.3.1.0 32kbit/s ADPCM to current ADPCM_DVI4*/
#define ORG_ADPCM_DVI4_TRANSF2_ADPCM_DVI4 0x63

/* transform Hi3510_VSSDK_V1.3.1.0 16kbit/s G.726 to current G726_16KBPS*/
#define ORG_G726_16KBPS_TRANSF2_G726_16KBPS 0x64

/* Type of the Hisilicon voice frame */
#define HI_NORMAL_FRAME           0x01 /* Voice frame */

/* Definition of the return value */
#define HI_SUCCESS                 0x00 /* Successful */

/* Definition of the error code prefix */
#define HI_ERR_VOICE_PREFIX       0xA1220000
```



```
/* Definition of the error code */
#define HI_ERR_VOICE_ENC_TYPE (HI_ERR_VOICE_PREFIX | 0x0001)
#define HI_ERR_VOICE_ENC_FRAME_SIZE (HI_ERR_VOICE_PREFIX | 0x0002)
#define HI_ERR_VOICE_DEC_TYPE (HI_ERR_VOICE_PREFIX | 0x0011)
#define HI_ERR_VOICE_DEC_FRAME_SIZE (HI_ERR_VOICE_PREFIX | 0x0012)
#define HI_ERR_VOICE_DEC_FRAME_TYPE (HI_ERR_VOICE_PREFIX | 0x0013)
#define HI_ERR_VOICE_INVALID_DEVICE (HI_ERR_VOICE_PREFIX | 0x0101)
#define HI_ERR_VOICE_INVALID_INBUF (HI_ERR_VOICE_PREFIX | 0x0102)
#define HI_ERR_VOICE_INVALID_OUTBUF (HI_ERR_VOICE_PREFIX | 0x0103)
#define HI_ERR_VOICE_TRANS_DEVICE (HI_ERR_VOICE_PREFIX | 0x1001)
#define HI_ERR_VOICE_TRANS_TYPE (HI_ERR_VOICE_PREFIX | 0x1002)
```

## hiVOICE\_G711\_STATE\_S

### [Description]

State structure for the G711 encoder or decoder.

### [Definition]

```
/* State for G711 encoder and decoder */
typedef struct hiVOICE_G711_STATE
{
    HI_S32 G711StateBuf[2];
} hiVOICE_G711_STATE_S;
```

### [Note]

The state structure for the G711 encoder or decoder is created independently.

## hiVOICE\_ADPCM\_STATE\_S

### [Description]

State structure for the ADPCM encoder or decoder.

### [Definition]

```
/* State for ADPCM encoder and decoder */
typedef struct hiVOICE_ADPCM_STATE
{
    HI_S32 AdpcmStateBuf[4];
} hiVOICE_ADPCM_STATE_S;
```

### [Note]

The state structure for the ADPCM encoder or decoder is created independently.

## hiVOICE\_G726\_STATE\_S

### [Description]



State structure for the G.726 encoder or decoder.

[Definition]

```
/* State for G726 encoder and decoder */  
typedef struct hiVOICE_G726_STATE  
{  
    HI_S32 G726StateBuf[32];  
} hiVOICE_G726_STATE_S;
```

[Note]

The state structure for the G.726 encoder or decoder is created independently.

## hiVOICE\_TRANSCODE\_STATE\_S

[Description]

State structure used to transcode the G711A, G711U, ADPCM\_DVI4, and G726\_16KBPS codes from the Hi3510\_VSSDK\_V1.3.1.0 versions to the later versions.

[Definition]

```
/* State for TransCode */  
typedef struct hiVOICE_TRANSCODE_STATE  
{  
    HI_S32 TransStateBuf[0x400];  
} hiVOICE_TRANSCODE_STATE_S;
```

[Note]

None.

## 3.2 Error Codes

[Table 3-1](#) lists the error codes in the document.

**Table 3-1** Error codes

| Error Code                  | Description   |
|-----------------------------|---|
| HI_ERR_VOICE_ENC_TYPE       | The encoder type is invalid.                          |
| HI_ERR_VOICE_ENC_FRAME_SIZE | The length of the data frame is invalid.              |
| HI_ERR_VOICE_DEC_TYPE       | The decoder type is invalid.                          |
| HI_ERR_VOICE_DEC_FRAME_SIZE | The length of the decoded data is invalid.            |
| HI_ERR_VOICE_DEC_FRAME_TYPE | The type of the audio frame is invalid.               |
| HI_ERR_VOICE_INVALID_DEVICE | The pointer to the encoder or the decoder is invalid. |
| HI_ERR_VOICE_INVALID_INBUF  | The input buffer is invalid.                          |





| Error Code                  | Description                               |
|-----------------------------|---|
| HI_ERR_VOICE_INVALID_OUTBUF | The output buffer is invalid.             |
| HI_ERR_VOICE_TRANS_TYPE     | The transcoder type is invalid.           |
| HI_ERR_VOICE_TRANS_DEVICE   | The pointer to the transcoder is invalid. |



# A Acronyms and Abbreviations

---

## A

**ADPCM** Adaptive Differential Pulse Code Modulation

## P

**PCM** Pulse Code Modulation