

# 快商通声纹技术文档

**Solutions with Voice Biometrics**

---

## 快商通声纹引擎 XVPR-WIN32-C 版本 技术开发文档

[厦门快商通信息技术有限公司](#)

[快商（厦门）软件科技有限公司](#)

文档版本	更新日志	更新时间	负责人
V1.0	1. 初始化文档	2013 年 07 月 05 日	张文斌
V1.1	1. 新增 VAD 实时检测接口 2. 新增语音内容校验接口	2014 年 1 月 02 日	李稀敏

## 目 录

1. 基本介绍.....	5
1.1 引擎简介.....	5
1.2 术语.....	5
1.2.1 声纹识别.....	5
1.2.2 声纹确认.....	5
1.2.3 声纹辨认.....	6
1.2.4 拒识.....	6
1.2.5 自识别.....	6
1.2.6 闯入识别.....	6
1.3 引擎适用平台.....	6
1.4 引擎适用开发语言.....	6
1.5 SDK 文件目录.....	7
1.6 使用注意事项.....	7
2. 声纹引擎接口.....	8
2.1 全局初始化接口.....	8
2.2 引擎信息获取接口.....	8
2.3 引擎资源释放接口.....	8
2.4 设置引擎参数.....	8
2.5 获取引擎句柄.....	9
2.6 创建说话人信息.....	9
2.7 删除说话人信息.....	9
2.8 获取说话人信息.....	10
2.9 添加说话人语音.....	10
2.10 删除说话人语音.....	10
2.11 保留最新注册语音.....	11
2.12 注册说话人声纹模型.....	11
2.13 更新说话人声纹模型.....	11
2.14 说话人声纹确认.....	11
2.15 说话人声纹辨认.....	12
3. 端点检测引擎接口.....	12
3.1 获取端点检测句柄.....	12
3.2 离线有效语音检测.....	13
3.3 实时语音端点检测.....	13
3.4 重置端点检测句柄.....	13
4. 引擎辅助接口.....	14
4.1 语音数据读取接口.....	14
4.2 语音数据写入接口.....	14
4.3 内存释放接口.....	14
5. 数据结构体说明.....	15
5.1 说话人信息结构体.....	15
5.2 识别结果结构体.....	15
6. 附件.....	16

附件一：函数返回值枚举类型.....	16
附件二：引擎参数说明.....	16

# 1. 基本介绍

## 1.1 引擎简介

**XVPR-WIN32-C 声纹识别引擎开发包是一个针对 WIN32 平台的 C 接口风格的开发包。**

声纹识别（又称说话人识别），是一项根据语音波形中反映说话人生理和行为特征语音参数来自动识别说话人身份的技术。声纹识别具有两个明显的优势：其一，非接触式识别，采集自然、易被接受；其二，麦克风作为声纹采集设备，成本低廉。这奠定了声纹识别被大规模普及推广的基础。快商通声纹识别开发平台基于全球领先的声纹识别引擎，为电话银行身份认证、司法社区矫正、社保生存验证、远程教育用户身份认证等应用保驾护航。平台具有以下特性：

- 支持文本相关、文本无关等多种模式
- 支持电话应用环境和桌面应用环境
- 支持说话人确认和说话人辨认
- 精准识别、稳定可靠
- 便捷、高效地集成

## 1.2 术语

### 1.2.1 声纹识别

生物识别技术的一种，是一项根据语音波形中反映说话人生理和行为特征的语音参数，自动识别说话人身份的技术。也称为说话人识别，有两类，即说话人辨认和说话人确认。不同的任务和应用会使用不同的声纹识别技术，如缩小刑侦范围时可能需要辨认技术，而银行交易时则需要确认技术。

### 1.2.2 声纹确认

也称为说话人确认，是声纹识别的一种。给定一个语音样本和一个说话人的声纹模型，声纹确认的任务是判断该语音样本是否就是该说话人所说。该说话人通常也称为“宣称的说

话人”。声纹确认是个“二值判别”问题，它的结果只有两种，或是肯定(即接受)，或是否定(即拒绝)。

### 1.2.3 声纹辨认

也称为说话人辨认，是声纹识别的一种。给定一个语音样本和一组说话人的声纹模型，声纹辨认的任务是判断该语音样本是哪个说话人所说的。因此，声纹辨认是一个“多选一”的问题。声纹辨认可分为闭集辨认或开集辨认。

快商通声纹引擎支持开集和闭集两种模式。

### 1.2.4 拒识

当无法给出一个识别结果或对识别结果没有足够把握时，识别器不输出任何识别结果，这称为拒识。在开集声纹辨认和声纹确认中，都存在拒识判别。

### 1.2.5 自识别

自识别指的是用识别语音和与识别语音话者身份相同的声纹模型进行声纹识别。

### 1.2.6 闯入识别

闯入识别指的是用识别语音和与识别语音话者身份不同的声纹模型进行声纹识别。

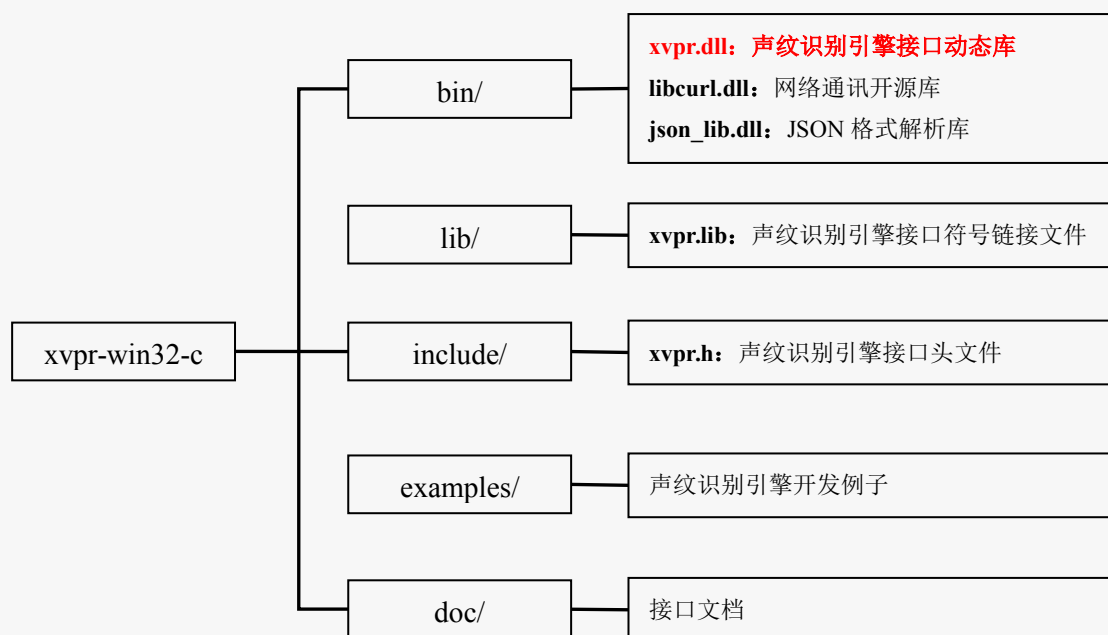
## 1.3 引擎适用平台

WIN-32，XP 及以上版本。

## 1.4 引擎适用开发语言

C/C++/C#/Delphi 等。

## 1.5 SDK 文件目录



## 1.6 使用注意事项

有以下几点需要注意：

- 本开发包是在快商通声纹识别核心服务上封装的一组 C 语言接口，需要先安装快商通声纹识别核心服务，才能正常调用、执行；
- 运行时，请将 `bin` 目录下所有的 `dll` 文件全部拷贝至 `exe` 所在的目录。

## 2. 声纹引擎接口

### 2.1 全局初始化接口

接口原型：

```
XVPR_CODE xvpr_global_init();
```

接口说明：

该接口用于初始化引擎资源，全局只可调用一次，而且必须调用。

参数说明：

无

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

### 2.2 引擎信息获取接口

接口原型：

```
const char *xvpr_version();
```

接口说明：

该接口用于获取引擎版本信息。

参数说明：

无

返回值说明：

版本信息字符串。

### 2.3 引擎资源释放接口

接口原型：

```
void xvpr_global_release();
```

接口说明：

该接口用于释放引擎所占用的资源。

参数说明：

无

返回值说明：

无

### 2.4 设置引擎参数

接口原型：

```
XVPR_CODE xvpr_global_setparam(const char *key, const char *param);
```



接口说明：

该接口用于设置引擎参数，包括服务器地址、端口、API密钥、引擎版本号等。

参数说明：

**key:** 参数关键字

**param:** 参数值

详细参数说明请参考【[附件二](#)】。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.5 获取引擎句柄

接口原型：

```
XVPR *xvpr_client_init(const char *key, const char *param);
```

接口说明：

该接口用于设置引擎参数，包括服务器地址、端口、API密钥、引擎版本号等。

参数说明：

**key:** 说话人群组号，是一个由 4 到 32 位字母或数字组成的字符串。

返回值说明：

返回引擎句柄，该句柄不能多线程共享，每一个线程中需要单独获取引擎句柄。

## 2.6 创建说话人信息

接口原型：

```
XVPR_CODE xvpr_create_person(XVPR *handle, const char *name);
```

接口说明：

该接口用于创建说话人信息。

参数说明：

**handle:** 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name:** 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.7 删除说话人信息

接口原型：

```
XVPR_CODE xvpr_remove_person(XVPR *handle, const char *name);
```

接口说明：

该接口用于删除说话人信息。

参数说明：

**handle:** 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name:** 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.8 获取说话人信息

接口原型：

```
XVPR_CODE xvpr_get_info(XVPR *handle, const char *name, xvpr_person_info &info);
```

接口说明：

该接口用于获取说话人信息。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name**: 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

**info**: 返回说话人信息。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.9 添加说话人语音

接口原型：

```
XVPR_CODE xvpr_add_speech(XVPR *handle, const char *name, const char *paramlist,  
short *stream, size_t buf_len, char *checksum);
```

接口说明：

该接口用于添加说话人语音。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name**: 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

**paramlist**: 用于设置语音参数。

**stream**: 语音流。

**buf\_len**: 语音采样点数（语音数组长度）。

**checksum**: 返回处理过后的语音的 MD5 值。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.10 删除说话人语音

接口原型：

```
XVPR_CODE xvpr_remove_speech(XVPR *handle, char *checksum);
```

接口说明：

该接口用于删除说话人语音。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**checksum**: 待删除语音的 MD5 值。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.11 保留最新注册语音

接口原型：

```
XVPR_CODE xvpr_reserve_speeches(XVPR *handle, const char *name);
```

接口说明：

该接口用于保留说话人最新的注册语音，以便删除较老的语音。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name**: 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.12 注册说话人声纹模型

接口原型：

```
XVPR_CODE xvpr_register_person(XVPR *handle, const char *name);
```

接口说明：

该接口用于注册说话人声纹模型，必须在上传3条及以上（建议3条）语音之后调用。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name**: 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.13 更新说话人声纹模型

接口原型：

```
XVPR_CODE xvpr_update_person(XVPR *handle, const char *name);
```

接口说明：

该接口用于更新说话人声纹模型，必须在上传新的语音之后调用。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name**: 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.14 说话人声纹确认

接口原型：

```
XVPR_CODE xvpr_verify_person(XVPR *handle, const char *name, const char *paramlist,
```

```
short *stream, size_t buf_len, xvpr_result &res);
```

接口说明：

该接口用于说话人声纹确认。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**name**: 说话人用户名，由 4 到 32 位字母或数字组成的字符串。

**paramlist**: 用于设置语音参数。

**stream**: 语音流。

**buf\_len**: 语音采样点数（语音数组长度）。

**res**: 返回声纹确认结果。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 2.15 说话人声纹辨认

接口原型：

```
XVPR_CODE xvpr_identify_person(XVPR *handle, const char *paramlist, short *stream,  
size_t buf_len, xvpr_result &res);
```

接口说明：

该接口用于说话人声纹辨认。

参数说明：

**handle**: 声纹识别引擎句柄，通过 `xvpr_client_init` 接口获取。

**paramlist**: 用于设置语音参数。

**stream**: 语音流。

**buf\_len**: 语音采样点数（语音数组长度）。

**res**: 返回声纹辨认结果。

返回值说明：

调用状态值，详细状态值说明请参考【[附件一](#)】。

## 3. 端点检测引擎接口

### 3.1 获取端点检测句柄

接口原型：

```
XVAD *xvpr_vad_create_vad(size_t sr);
```

接口说明：

该接口用于创建端点检测引擎实例。注意，该句柄不能在多线程间共享，必须为每个线程独立分配句柄。

参数说明：

**sr**: 语音采样率。

返回值说明：

返回端点检测实例句柄。

## 3.2 离线有效语音检测

接口原型：

```
size_t xvpr_vad_detect_valid_speech(XVPR *&handle, short *input, size_t length, short *output);
```

接口说明：

该接口用于对原始语音进行离线有效语音检测，返回有效语音。

参数说明：

**handle**: 端点检测引擎句柄，通过 `xvpr_vad_create_vad` 接口获取。

**input**: 原始语音流。

**length**: 原始语音采样点数（语音数组长度）。

**output**: 检测后得到的有效语音流。

返回值说明：

返回语音采样点数（语音数组长度）。

## 3.3 实时语音端点检测

接口原型：

```
int xvpr_vad_is_speaking(XVPR *&handle, short *stream, size_t length);
```

接口说明：

该接口用于实时语音端点检测。

参数说明：

**handle**: 端点检测引擎句柄，通过 `xvpr_vad_create_vad` 接口获取。

**stream**: 实时语音流。

**buf\_len**: 原始语音采样点数（语音数组长度）。

返回值说明：

若返回值为 0，则表示还在说话；若为 1，则表示停止说话了。

## 3.4 重置端点检测句柄

接口原型：

```
void xvpr_vad_is_speaking(XVPR *&handle);
```

接口说明：

该接口用于重置端点检测句柄。

参数说明：

**handle**: 端点检测引擎句柄，通过 `xvpr_vad_create_vad` 接口获取。

返回值说明：

无。

## 4. 引擎辅助接口

### 4.1 语音数据读取接口

接口原型：

```
size_t xvpr_aid_read_buffer(const char *path, size_t *skip_bits, short *&buffer);
```

接口说明：

该接口用于从语音文件中读取语音数据（目前仅支持PCM/RAW格式的语音文件）。

参数说明：

path: 语音文件路径。

skip\_bits: 需要跳过头部的字节数。

buffer: 返回语音数据。内存由接口内部申请，并且需要开发者调用【释放内存】的接口进行释放。

返回值说明：

返回读取到的语音数组长度。

### 4.2 语音数据写入接口

接口原型：

```
int xvpr_aid_write_buffer(const char *path, short *buffer, size_t *buf_len);
```

接口说明：

该接口用于从语音文件中读取语音数据（目前仅支持PCM/RAW格式的语音文件）。

参数说明：

path: 语音文件路径。

buffer: 语音数据缓冲。

buf\_len: 语音采样点个数。

返回值说明：

若写入成功，则返回 0；否则，返回-1。

### 4.3 内存释放接口

接口原型：

```
void xvpr_aid_release_buffer(short *&buffer);
```

接口说明：

该接口用于释放引擎内部为开发者申请的内存空间，比如【语音数据读取接口】。

参数说明：

buffer: 语音数据缓冲。

返回值说明：

若写入成功，则返回 0；否则，返回-1。

## 5. 数据结构体说明

### 5.1 说话人信息结构体

结构体原型：

```
struct xvpr_person_info {  
    char collection[32];    /* 说话人群组 */  
    char name[32];         /* 说话人名字 */  
    char tag[32];          /* 说话人标签 */  
    bool flag;             /* 登记状态 */  
}
```

用途：

用于返回说话人信息。

### 5.2 识别结果结构体

结构体原型：

```
struct xvpr_result {  
    char collection[32];    /* 说话人群组 */  
    char name[32];         /* 说话人名字 */  
    float similarity;       /* 相似度 */  
    char gender;           /* 说话人性别 */  
}
```

用途：

用于返回说话人确认或辨认结果。

## 6. 附件

### 附件一：函数返回值枚举类型

值	说明
<code>XVPR_CODE_SUCCESS</code>	调用成功
<code>XVPR_CODE_FAIL</code>	调用失败
<code>XVPR_CODE_CONNECT_ERR</code>	网络连接错误
<code>XVPR_CODE_PARAM_ERR</code>	参数错误
<code>XVPR_CODE_NOT_FOUND</code>	对象未找到

### 附件二：引擎参数说明

参数关键字	参数值	说明
<code>host</code>	例如：127.0.0.1	声纹识别服务器的 IP 地址
<code>port</code>	例如：82	声纹核心服务器端口，默认 82 端口
<code>accesskey</code>	32 位接口密钥	声纹服务器接口密钥，由快商通提供
<code>secretkey</code>	32 位安全密钥	声纹服务器安全密钥，由快商通提供
<code>version</code>	版本号	声纹服务器版本号

调用示例：

```
xvpr_global_setparam("host", "127.0.0.1");  
xvpr_global_setparam("port", "82");  
xvpr_global_setparam("accesskey", "1ee0d9b37g8d92a1555abc85e0b7074e");  
xvpr_global_setparam("secretkey", "ks90dsda7g8d92a150i4sbc85e0dasdc2");  
xvpr_global_setparam("version", "1");
```