****

**Open AVL SDK**

**Interface Specification Manual**

www.antiy.com

**contents**

[1 Function introduction 2](#_Toc70686330)

[2 Open AVL SDK Development Guide 2](#_Toc70686331)

[2.1 Interface Implemented by Users 2](#_Toc70686332)

[Callback Function P\_OBJ\_DIPOSER\_CALLBACK 2](#_Toc70686333)

[Callback Function P\_QUERY\_CONTINUE\_CALLBACK 3](#_Toc70686334)

[2.2 User Calling Steps 4](#_Toc70686335)

[2.3 The List of Exported Functions by Open AVL SDK 5](#_Toc70686336)

[2.4 Definition of Structure of AVL SDK 5](#_Toc70686337)

[2.4.1 Definition of OBJ\_PROVIDER Struct 5](#_Toc70686338)

[2.4.2 Definition of OBJ\_DISPOSER Struct 6](#_Toc70686339)

[2.5 Open AVLSDK data definition used in read / write configuration 6](#_Toc70686340)

[2.5.1 Data Definition in Data Report (AVL\_SDK\_QueryReport Function) 7](#_Toc70686341)

[2.5.2 Open AVLSDK Configuration of the Engine Working Environment 7](#_Toc70686342)

[2.5.3 Open AVLSDK configuration-related information 8](#_Toc70686343)

[2.6 Open AVLSDK API Reference 14](#_Toc70686344)

[3 API Reference 14](#_Toc70686345)

[3.1 AVL\_SDK\_CreateInstance 14](#_Toc70686346)

[3.2 AVL\_SDK\_LoadConfigFile 15](#_Toc70686347)

[3.3 AVL\_SDK\_SetConfigInt 16](#_Toc70686348)

[3.4 AVL\_SDK\_SetConfigString 16](#_Toc70686349)

[3.5 AVL\_SDK\_GetConfigInt 17](#_Toc70686350)

[3.6 AVL\_SDK\_GetConfigString 18](#_Toc70686351)

[3.7 AVL\_SDK\_InitInstance 19](#_Toc70686352)

[3.8 AVL\_SDK\_Scan 20](#_Toc70686353)

[3.9 AVL\_SDK\_Release 21](#_Toc70686354)

[3.10 AVL\_SDK\_QueryReportInt 22](#_Toc70686355)

[3.11 AVL\_SDK\_ QueryReportString 23](#_Toc70686356)

[3.12 AVL\_SDK\_GetLicenseExpDate 24](#_Toc70686357)

[3.13 AVL\_SDK\_ GetCurVersion 25](#_Toc70686358)

# Function introduction

Based on avlsdk, Open AVL SDK is a lightweight transformation to form open AVL SDK, which provides efficient, fast and professional security protection capability interface for cooperative developers, and a free solution with anti-virus capability for security products or services.

# Open AVL SDK Development Guide

## Interface Implemented by Users

Users need to implement P\_OBJ\_DISPOSER\_CALLBACK, and P\_QUERY\_CONTINUE\_CALLBACK call-back functions. When users call AVL\_SDK\_Scan for detection, the pointer of the call-back functions will be sent. The engine will send the detection results to users via calling P\_OBJ\_DISPOSER\_CALLBACK function.

### Callback Function P\_OBJ\_DIPOSER\_CALLBACK

The engine will send the current detection results to users.

**typedef long (\*P\_OBJ\_DISPOSER\_CALLBACK)(**

**/\*[in]\*/ P\_OBJ\_PROVIDER p\_op,**

**/\*[in]\*/ void \*p\_rpt\_handle,**

**/\*[in]\*/ void \*p\_param**

**);**

Parameter:

p\_op

It returns the object detection OBJ\_PROVIDER pointer.

p\_rpt\_handle

It returns the object detection results to the handle pointer, which is used when calling the AVL\_SDK\_QueryReport function; and the current scan results can be obtained.

p\_param

It is the parameter pointer transferred by user.

Results Returned:

Meaningless currently

Description：

When the engine finishes detecting an object, it will call the callback function to send the detection results to the user. The user can call AVL\_SDK\_QueryReport interface function through the p\_rpt\_handle handle pointer to obtain any information. p\_param is the parameter pointer that the user needs to pass to the callback function, and the engine will send it to the callback function wholly intact.

eg:

reference Demo/main.c func\_long\_get\_rslt\_callback

### Callback Function P\_QUERY\_CONTINUE\_CALLBACK

The engine inquires the user whether it should continue to work via this callback function.

**typedef long (\*P\_QUERY\_CONTINUE\_CALLBACK)(**

**/\*[in]\*/ void \*p\_param**

**);**

Parameter:

p\_param

The parameter pointer that the user sends to this callback function

Results Returned：

OD\_CONTINUE or OD\_ABORT

Description：

Since the user may need to terminate the scan of the engine any time, the engine will inquire the user whether it should continue to work before the start of detection so that the user can terminate the scan any time. p\_param is the parameter pointer that the user needs to pass to the callback function, and the engine will send it to the callback function wholly intact.

eg:

long func\_long\_query\_continue\_callback(void \*p\_param)

{

// This is the code sample, so it returns unconditionally. Users can modify according to the condition.

return OD\_CONTINUE;

}

or reference Demo/main.c

## User Calling Steps

1. Implement callback function P\_OBJ\_DISPOSER\_CALLBACK and P\_QUERY\_CONTINUE\_CALLBACK;
2. Load the main modules of the engine (AVLSDK.so) and export the API the engine provides;

1）Configure module path: /home/engine/AVLSDK.so

*func\_long\_loadLibrary(” /home/engine/AVLSDK.so”, p\_engine)*

1. Create a new engine instance ([AVL\_SDK\_CreateInstance](#_AVLV_SDK_CreateInstance));
2. Load the configuration template files ([AVL\_SDK\_LoadConfigFile](#_AVLV_SDK_LoadConfigFile))
3. Configure the engine switch ([AVL\_SDK\_SetConfigInt](#_AVLV_SDK_SetConfigInt)\[AVL\_SDK\_SetConfigString](#_AVLV_SDK_SetConfigString))

egg:

1. Configure engine library path :/home/engine/Data

[*AVL\_SDK\_SetConfigString*](#_AVLV_SDK_SetConfigString)*(p\_engine,CFG\_STR\_DATA\_PATH,”/home/engine/*D*ata”)*

1. Configure module path :/home/engine/Module

[*AVL\_SDK\_SetConfigString*](#_AVLV_SDK_SetConfigString)*(p\_engine,CFG\_STR\_MODULE\_PATH,”/home/engine/Module”)*

1. Configure SDK Lincense.alf path: /home/engine/Lincense/License.alf

[*AVL\_SDK\_SetConfigString*](#_AVLV_SDK_SetConfigString)*(p\_engine, CFG\_STR\_LICENSE\_PATH,” /home/engine/Lincense/License.alf”)*

1. Initialize the engineinstance ([AVL\_SDK\_InitInstance](#_AVLV_SDK_InitInstance))；
2. Scan([AVL\_SDK\_Scan](#_AVLV_SDK_Scan))；
3. Set the query SDK version (AVL\_SDK\_GetCurVersion)

Note: this item can be set as needed

1. Set the valid time of the query license (AVL\_SDK\_GetLicenseExpDate)

Note: this item can be set as needed

1. Release the engine instance when it is no longer needed ([AVL\_SDK\_Release](#_AVLV_SDK_Release))

## The List of Exported Functions by Open AVL SDK

Refer to [API Reference](#_API参考) for detailed information of functions.

List 2-1 Open AVL SDK Exported Function

|  |  |
| --- | --- |
| Function Name | Description |
| AVL\_SDK\_CreateInstance | AVL SDK creates new engine instances |
| AVL\_SDK\_LoadConfigFile | Load the configuration files |
| AVL\_SDK\_SetConfigInt | AVL SDK configuration switch |
| AVL\_SDK\_SetConfigString | AVL SDK configuration switch |
| AVL\_SDK\_GetConfigInt | AVL SDK receives the current configuration |
| AVL\_SDK\_GetConfigString | AVL SDK receives the current configuration |
| AVL\_SDK\_InitInstance | Initialize the engine instance based on the configuration information |
| AVL\_SDK\_Scan | Scan function |
| AVL\_SDK\_QueryReportInt | Inquire the detection results of the object instances |
| AVL\_SDK\_QueryReportStr | Inquire the detection results of the object instances |
| AVL\_SDK\_Release | Release the engine instance |
| AVL\_SDK\_GetCurVersion | Gets the version of the current engine |
| AVL\_SDK\_GetLicenseExpDate | Gets the date of the License file for the current engine |

## Definition of Structure of AVL SDK

This section mainly introduces the structure definition and related data macro definition of Open AVLSDK

### Definition of OBJ\_PROVIDER Struct

The OBJ-PROVIDER Struct is mainly used to alternate data between users and engines, users need to provide the data buffer and the buffer size to be scanned by engines, and engines will also send the new data to the callback function that users had registered through this Struct. For the definition of OBJ\_PROVIDER Strut, please refer to file engine.h.

* **obj\_ver:**

It is the version of this struct.

* **obj\_type:**

Data type (Reserved field currently).

* **evro\_type:**

The type of engine work environment. For more details, please refer to the related instruction of environment type.

* **buf:**

Data buffer pointer.

* **size:**

The size of valid data in buffer

* **obj\_des:**

Data description information, users need to store the complete path information of data here.

### Definition of OBJ\_DISPOSER Struct

The OBJ-DISPOSER Struct is mainly used by users to pass the disposal callback functions to engines. Engines need to be interacted (such as noticing scan suspend, obtaining scan result and so on) with users during the working period, so engines pass the interactive interface function through this struct. For the definition details of OBJ-DISPOSER Struct, please refer to file engine.h.

* **rpt\_callback:**

It is the callback function pointer that receives the detection result report that engines send to users, you can find the definition at P\_OBJ\_DIPOSER\_CALLBACK

* **p\_rpt\_param:**

The perimeter pointer that users pass to the callback function rpt\_callback

* **query\_continue\_callback:**

It is the callback function pointer that replies whether the engine suspends, you can find the definition at P\_QUERY\_CONTINUE\_CALLBACK.

* **p\_qc\_param:**

The perimeter pointer that users pass to the callback function query\_continue\_callback

## Open AVLSDK data definition used in read / write configuration

Open AVLSDK configuration options are classified based on usage types as follows

### Data Definition in Data Report (AVL\_SDK\_QueryReport Function)

|  |  |
| --- | --- |
| Definition | Description |
| RPT\_IDX\_OBJ\_DESCRIPTION | Description of the data object |
| RPT\_IDX\_MALWARE\_ID | Get the malware ID detected by the engine |
| RPT\_IDX\_ANALYSER | Get the name of the detection module reporting the MALWAE\_ID |
| RPT\_IDX\_CLOUD\_MALWARENAME\_ID | Get malware name by cloud detection |

### Open AVLSDK Configuration of the Engine Working Environment

Table 2-2 Open AVLSDK Configuration of the Engine Working Environment

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Value Range | Description |
| **CFG\_STR\_MODULE\_PATH** | Character | Depends on the environment | This is to demonstrate path of the engine sub-module. Different versions of modules can be placed in different paths to allow different engine instances to load. |
| **CFG\_STR\_DATA\_PATH** | Character | Depends on the environment | This is to demonstrate path of the virus database. Different versions of virus database can be placed in different paths to allow different engine instances to load, which consequently realizes the hot restart of the engine update database. |
| **CFG\_STR\_LICENSE\_PATH** | Character | Depends on the environment | The full path for the engine to be used for the License file, not the path to the directory where the License file is located |
| **CFG\_STR\_LOG\_PATH** | Character | Depends on the environment | This is to demonstrate path of the engine Log. In this process, the reading/writing privileges are necessary. |

### Open AVLSDK configuration-related information

Use AVL\_SDK\_SetConfigInt/AVL\_SDK\_SetConfigString function interface configuration module,If it is not set, the default value will be used。

Table 2-3 Open AVLSDK Module read / write config

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable | Type | Extent | Description | Default |
| **CFG\_FLAG\_CLOUD\_DETECT\_ENABLE** | Boolean | 0 or 1 | This configuration option is the runtime switch, which is different from the loading-time switch. It can still modify its value at any time after the engine initialization case in order to reach the control demands.  0 no load. 1 load | 1 |
| **CFG\_INT\_CLOUD\_CONNECT\_TIMEOUT** | Integer | 0 ~ 0xffffffff | This configuration option is mainly used to set the timeout of connecting the cloud server, which is invalid after initialization. | 3 |
| **CFG\_INT\_CLOUD\_DETECT\_TIMEOUT** | Integer | 0 ~ 0xffffffff | This configuration option is mainly used to set the timeout of cloud server when detecting single file, which is invalid after initialization.  （unit：s） | 5 |
| **CFG\_INT\_CLOUD\_MAX\_SCAN\_FILE\_SIZE** | Integer | 0 ~ 0xffffffff | This configuration option mainly sets the maximum file size that can be detected. This configuration option can still modify its value at any time after the engine initializes the instance to meet the control requirements.（unit：byte） | 1 \* 1024 \* 1024 |
| **CFG\_INT\_CLOUD\_CACHE\_MAX\_RECORD\_SIZE** | Integer | 0 ~ 0xffffffff | Cache the MD5 detection results of the latest query, and the next detection is more efficient than MD5 | 10000 |
| **CFG\_INT\_CLOUD\_CACHE\_VALID\_HOURS** | Integer | 0 ~ 0xffffffff | Cache save time（unit：h） | 24 |

## Open AVLSDK API Reference

Open AVLSDK API reference include two types: success and exception. The specific meaning is shown in table 2-4 to table 2-5

Table 2-4 success

|  |  |  |
| --- | --- | --- |
| Macro | Value | Description |
| ERR\_SUCCESS | 0 | success |

Table 2-5 return error

|  |  |  |
| --- | --- | --- |
| 宏 | Value | Description |
| ERR\_UNKNOWN | -1 | unknown error |
| ERR\_INCORRECT\_PARAM | -2 | parameter error |
| ERR\_NO\_ENOUGH\_MEM | -3 | insufficient memory |
| ERR\_BAD\_LICENSE | -4 | set License fail |
| ERR\_BAD\_AUTH\_CODE | -5 | Author Code error |
| ERR\_BAD\_ENGINE\_VER | -6 | version error |
| ERR\_BAD\_DATABASE | -7 | data error |
| ERR\_INIT\_COMPONENT\_FAILED | -8 | initialization failed |
| ERR\_MISS\_IMPORTANT\_MOD\_FILE | -9 | lack module |
| ERR\_MISS\_IMPORTANT\_DATA\_FILE | -10 | lack data file |
| ERR\_BAD\_WORKING\_PATH | -11 | working path error |
| ERR\_LOAD\_CONFIG | -12 | load config error |
| ERR\_SET\_CONFIG | -13 | set parameter error |
| ERR\_GET\_CONFIG | -14 | get parameter error |

# API Reference

This chapter mainly introduces the call interfaces of Open AVL SDK

## AVL\_SDK\_CreateInstance

AVL SDK engine creates a new engine instance function, which is called before loading the configuration options.

***long AVL\_SDK\_CreateInstance（***

***/\*[out]\*/ void \*\*ppEngine***

***）;***

Parameter：

ppEngine

The newly created engine instance; the parameter is sent to the pointer address, and the engine instance pointer is sent out and can’t be set to NULL.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table。

Description：

Using this interface to load configuration template can load relevant configuration templates provided by the engine.

## AVL\_SDK\_LoadConfigFile

Open AVL SDK engine sets the data receiver, and the function is called before scanning.

***long AVL\_SDK\_LoadConfigFile（***

***/\*[in]\*/ void \*pEngine，***

***/\*[in]\*/ char \*szFilename***

***）;***

Parameter：

pEngine

It is the engine instance pointer, which can’t be set to NULL.

szFilename

Configure the full-path pointer of templates, which cannot be set to NULL.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

Using this interface to load configuration template can load relevant configuration templates provided by the engine.

## AVL\_SDK\_SetConfigInt

AVL SDK engine configuration function

***long AVL\_SDK\_SetConfigInt（***

***/\*[in]\*/ void \*pEngine，***

***/\*[in]\*/long CfgIdx ,***

***/\*[in]\*/long lValue***

***）;***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

CfgIdx

Engine configuration index, which is used to point out the switch needed to be configured, can’t be set to NULL. The details of configurable switches can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_conf\_idx.h.

lValue

Engine configuration switch value, has corresponding configuration values ​​for different configuration switches. The details of configuration values can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_conf\_idx.h.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

This interface is used to configure the integer as a configurable switch of a configurable value.

## AVL\_SDK\_SetConfigString

AVL SDK engine configuration function

***long AVL\_SDK\_SetConfigString（***

***/\*[in]\*/ void \*pEngine，***

***/\*[in]\*/long CfgIdx ,***

***/\*[in]\*/char\* lValue***

***）;***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

CfgIdx

Engine configuration item, which is used to point out the switch needed to be configured, can’t be set to NULL. The details of configurable switches can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_conf\_idx.h.pValue

lValue

Engine configuration switch value, has corresponding configuration values ​​for different configuration switches. The details of configuration values can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_conf\_idx.h.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

This interface is used to configure the string as a configurable switch of a configurable value.

## AVL\_SDK\_GetConfigInt

AVL SDK engine get the current configuration function (Integer).

***long AVL\_SDK\_GetConfigInt（***

***/\*[in]\*/ void \*pEngine，***

***/\*[in]\*/long CfgIdx ,***

***/\*[out]\*/long \*pValue ，***

***/\*[in]\*/long default\_val***

***）;***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

CfgIdx

Engine configuration item, which is used to point out the switch needed to be configured, can’t be set to NULL. The details of configurable switches can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_conf\_idx.h.

pValue

Used to obtain corresponding values of the configuration items

default\_val

p\_Value default, Used to set default value when getting configuration item value failed

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

This interface is used to obtain the current engine instance integer configuration values.

## AVL\_SDK\_GetConfigString

AVL SDK engine get the current configuration function (String).

***long AVL\_SDK\_GetConfigString（***

***/\*[in]\* void \*pEngine，***

***/\*[in]\*/ long CfgIdx ,***

***/\*[out]\*/char \*\*Buf,***

***/\*[out]\*/long \*BufLen,***

***/\*[in]\*/char \*Default\_Buf***

***）;***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

CfgIdx

Engine configuration item, which is used to point out the switch needed to be configured, can’t be set to NULL. The details of configurable switches can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_conf\_idx.h.

Buf

Used to store the obtained character strings

BufLen

Used to state the available space of Buf

Default\_Buf

Set the default value that cannot be obtained, usually set to NULL.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

This interface is used to obtain the current engine instance string configuration values.

## AVL\_SDK\_InitInstance

According to the configuration, AVL SDK engine initializes new engine instances, load the database and detection module. After initialized, the engine instance is available. This interface should be called after the engine instance having been created.

***long AVL\_SDK\_InitInstance（***

***/\*[in]\*/ void \*pEngine，***

***/\*[in]\*/ const void \*pVerificationCode,***

***）;***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

pVerificationCode

Engine authorization verification code.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

This interface starts to initialize the engine instance; however, AVL\_SDK\_CreateInstance is just used to allocate memory space. If the engine instance fails to be initialized, this interface will automatically call and release the interface of the engine instance.

## AVL\_SDK\_Scan

AVL SDK engine scanning function to scan the object to be processed.

***long AVL\_SDK\_Scan（***

***/\*[in]\*/ void \*pEngine，***

***/\*[in]\*/ P\_OBJ\_VECTOR\_PROVIDER pObj***

***/\*[in]\*/ P\_OBJ\_DISPOSER\_VECTOR\_CALLBACK vector\_callback，***

***/\*[in]\*/void \*p\_param***

***）;***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

pObj

Data (data provider) to be scanned, which needs users to fill its structure and send pointers to scanning function. Its structure is defined in detail in [OBJ\_VECTOR\_PROVIDER](#_数据提供者(OBJ_VECTOR_PROVIDER)结构体定义) structure definition.

P\_OBJ\_DISPOSER\_VECTOR\_CALLBACK

Object disposer structure pointer, which needs users to fill its structure and send pointers to scanning function. The engine will call P\_QUERY\_CONTINUE\_CALLBACK to query whether the user interrupts the scan or not, then call P\_QUERY\_CONTINUE\_CALLBACK to send the scanning result to users.

p\_param

Test result processing callback functionParameter。

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

This interface should be called after the engine instance has been initialized. More information of the two structures can be seen in relevant definitions.

## AVL\_SDK\_Release

Release AVL SDK engine instance, including the database and detection module etc.

***long AVL\_SDK\_Release（***

***/\*[in]\*/ void \*pEngine***

***）;***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

Release a specified engine instance, but without any effect on other engine instances in work.

## AVL\_SDK\_QueryReportInt

AVL SDKengine test results are reported to the query interface (Integer).

***long AVL\_SDK\_QueryReportInt(***

***/\*[in]\*/ void \*pEngine,***

***/\*[in]\*/ void \*pRptHandle,***

***/\*[in]\*/ unsigned long key,***

***/\*[out]\*/ long \*value***

***);***

pEngine

Engine instance pointer can’t be set to NULL.

pRptHanlde

Report handle pointer delivered to the user can’t be set to NULL.

key

Engine supports report item, its more details can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_rpt\_idx.h.

value

Buffer pointer provided by users used to save the corresponding results of report items.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

The engine instance of this interface can pass the corresponding parameters of the function P\_OBJ\_DISPOSER\_CALLBACK, which can send the report item handle pointer to the user.

## AVL\_SDK\_ QueryReportString

AVL SDK engine test results report query interface (String).

***long AVL\_SDK\_QueryReportString(***

***/\*[in]\*/ void \*pEngine,***

***/\*[in]\*/ void \*pRptHandle,***

***/\*[in]\*/ unsigned long key,***

***/\*[out]\*/ unsigned char \*\*value***

***);***

Parameter：

pEngine

Engine instance pointer can’t be set to NULL.

pRptHanlde

Report handle pointer delivered to the user can’t be set to NULL.

key

Engine supports report item, its more details can be seen in the file declarations or (where adding links to configuration switch entry) of AVLSDK\_rpt\_idx.h or ([2.5.1](#_数据报告（AVL_SDK_QueryReport函数)所用到的数据定义))

value

The pointer provided by the user to save the buffer pointer corresponding to the report item cannot be set to NULL。

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

The engine instance of this interface can pass the corresponding parameters of the function P\_OBJ\_DISPOSER\_CALLBACK, which can send the report item handle pointer to the user.

## AVL\_SDK\_GetLicenseExpDate

Gets the date of the License file for the current engine.

***long AVL\_SDK\_GetLicenseExpDate（***

***/\*[in]\*/ void \*pEngine,***

***/\*[in]\*/ unsigned char \*buf,***

***/\*[in]\*/ unsigned long len***

***）;***

Parameter：

pEngine

System function set pointer can not be set to NULL

buf

Used to save the license time of the buffer and it can not be set to NULL. The length of the buffer can not be less than 32 bytes.

len

Used to declare the length of the buffer.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

If the length of the buffer is not enough to save the engine version string, the engine will fill the buffer and will retain the last byte for the '\0'. At the same time, the function will not return with an exception, we suggest that this buffer size is not less than 128 bytes.

## AVL\_SDK\_ GetCurVersion

Gets the release version of the current engine.

***long AVL\_SDK\_GetCurVersion(***

***/\*[in]\*/ unsigned char \*buf,***

***/\*[in]\*/ unsigned long len***

***);***

Parameter：

buf

Used to save the engine version of the buffer and it can not be set to NULL. The length of the buffer can not be less than 32 bytes.

len

Used to declare the length of the buffer.

Results Returned：

Returned error codes, more detailed information can be seen 2.6 Open AVLSDK API Reference table

Description：

If the length of the buffer is not enough to save the engine version string, the engine will fill the buffer and will retain the last byte for the '\0'. At the same time, the function will not return with an exception, we suggest that this buffer size is not less than 128 bytes.