****

**AVL开放引擎接口规范手册**

反病毒引擎研发中心

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **文档信息** | | | | | |
| **文档名称** | AVL开放引擎接口规范手册 | | | | |
| **文档版本** |  | | **发布范围** | 公开 | |
| **编写人员** | 干超 | | **编写日期** | 2021/4/28 | |
| **文档摘要** |  | | | | |
| **修订记录** | | | | | |
| **修订日期** | **版本** | **说明** | | | **修订人** |
|  |  |  | | |  |

www.antiy.com

**目 录**

[1 功能简介 2](#_Toc70597936)

[2 AVL开放引擎开发向导 2](#_Toc70597937)

[2.1 用户需要实现的接口 2](#_Toc70597938)

[P\_OBJ\_DIPOSER\_CALLBACK回调函数 2](#_Toc70597939)

[P\_QUERY\_CONTINUE\_CALLBACK回调函数 3](#_Toc70597940)

[2.2 用户调用流程 3](#_Toc70597941)

[2.3 AVL 开放引擎导出函数列表 4](#_Toc70597942)

[2.4 引擎结构定义 5](#_Toc70597943)

[2.4.1 P\_ENGINE\_PARAM结构体定义 5](#_Toc70597944)

[2.4.2 数据提供者（OBJ\_PROVIDER）结构体定义 6](#_Toc70597945)

[2.4.3 数据处置者（OBJ\_DISPOSER）结构体定义 6](#_Toc70597946)

[2.5 开放引擎读写配置所用到的数据定义 7](#_Toc70597947)

[2.5.1 数据报告（AVL\_SDK\_QueryReport函数)所用到的数据定义） 7](#_Toc70597948)

[2.5.2 开放引擎工作环境配置 7](#_Toc70597949)

[2.5.3 开放引擎模块配置 8](#_Toc70597950)

[2.6 开放引擎API返回值 12](#_Toc70597951)

[3 API参考 12](#_Toc70597952)

[3.1 AVL\_SDK\_CreateInstance 13](#_Toc70597953)

[3.2 AVL\_SDK\_LoadConfigFile 13](#_Toc70597954)

[3.3 AVL\_SDK\_SetConfigInt 14](#_Toc70597955)

[3.4 AVL\_SDK\_SetConfigString 14](#_Toc70597956)

[3.5 AVL\_SDK\_GetConfigInt 15](#_Toc70597957)

[3.6 AVL\_SDK\_GetConfigString 16](#_Toc70597958)

[3.7 AVL\_SDK\_InitInstance 17](#_Toc70597959)

[3.8 AVL\_SDK\_Scan 18](#_Toc70597960)

[3.9 AVL\_SDK\_Release 19](#_Toc70597961)

[3.10 AVL\_SDK\_QueryReportInt 19](#_Toc70597962)

[3.11 AVL\_SDK\_ QueryReportString 20](#_Toc70597963)

[3.12 AVL\_SDK\_GetLicenseExpDate 21](#_Toc70597964)

[3.13 AVL\_SDK\_ GetCurVersion 22](#_Toc70597965)

**AVL开放引擎接口规范手册**

反病毒引擎研发中心

# 功能简介

Open AVL SDK是基于AVLSDK的基础上，进行轻量改造，形成Open AVL SDK，为合作开发者提供高效、快速和专业的安全防护能力接口，为安全类产品或服务提供反病毒能力的免费解决方案。

# AVL开放引擎开发向导

## 用户需要实现的接口

用户需实现P\_OBJ\_DISPOSER\_CALLBACK、P\_QUERY\_CONTINUE\_CALLBACK回调函数，在用户调用AVL\_SDK\_Scan进行检测时，将回调函数的指针传入，引擎通过调用P\_OBJ\_DISPOSER\_CALLBACK函数通知用户检测结果。

### P\_OBJ\_DIPOSER\_CALLBACK回调函数

将引擎的当前的检测结果通知用户。

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

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

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

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

**);**

参数：

p\_op

返回对象检测数据提供者指针。

p\_rpt\_handle

返回对象检测结果报告句柄指针，供调用AVL\_SDK\_QueryReport函数时使用，可获得当前的扫描结果。

p\_param

用户传递给此回调函数的参数指针。

返回结果：

目前无意义。

说明：

当引擎检测完一个对象之后，会调用此回调函数将检测结果通知给用户，用户可以通过p\_rpt\_handle句柄指针调用AVL\_SDK\_QueryReport接口，获取相关任何信息。其中，p\_param是用户需要传递给回调函数的参数指针，引擎会原封不动的交给回调函数。

### P\_QUERY\_CONTINUE\_CALLBACK回调函数

引擎通过此回调函数向用户询问引擎是否继续工作。

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

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

**);**

参数：

p\_param

用户传递给此回调函数的参数指针。

返回结果：

OD\_CONTINUE 或者 OD\_ABORT。

说明：

由于用户随时可能需要终止引擎的扫描动作，所以引擎会在每一次扫描开始前向用户询问是否继续工作，以便于用户可以随时终止扫描动作。其中，p\_param是用户需要传递给回调函数的参数指针，引擎会原封不动的交给回调函数。

## 用户调用流程

1. 实现P\_OBJ\_DISPOSER\_CALLBACK及P\_QUERY\_CONTINUE\_CALLBACK回调函数
2. 加载开放引擎主模块(AVLSDK.so)并导出开放引擎提供的API；

1）配置主模块路径 /home/engine/AVLSDK.so

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

1. 创建一个新的引擎实例([AVL\_SDK\_CreateInstance](#_AVLV_SDK_CreateInstance));
2. 加载配置模板([AVL\_SDK\_LoadConfigFile](#_AVLV_SDK_LoadConfigFile))
3. 配置开放引擎开关([AVL\_SDK\_SetConfigInt](#_AVLV_SDK_SetConfigInt)\[AVL\_SDK\_SetConfigString](#_AVLV_SDK_SetConfigString))，在加载配置模板上进行配置项修改及新增配置，如
4. 配置引擎库路径为/home/engine/Data

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

1. 配置模块路径为/home/engine/Module

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

1. 配置SDK Lincense.alf路径为/home/engine/Lincense/License.alf

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

1. 初始化开放引擎实例([AVL\_SDK\_InitInstance](#_AVLV_SDK_InitInstance))；
2. 扫描([AVL\_SDK\_Scan](#_AVLV_SDK_Scan))；
3. 设置查询sdk版本(AVL\_SDK\_GetCurVersion)

注：此项可视需要设置

1. 设置查询License有效时间(AVL\_SDK\_GetLicenseExpDate)

注：此项可视需要设置

1. 不再需要开放引擎实例工作时，将其释放([AVL\_SDK\_Release](#_AVLV_SDK_Release))

## AVL 开放引擎导出函数列表

函数的详细信息请参见[API参考](#_API参考)

表 2-1 AVL开放引擎导出函数列表

|  |  |
| --- | --- |
| 函数名称 | 说明 |
| AVL\_SDK\_CreateInstance | AVL SDK创建新引擎实例 |
| AVL\_SDK\_LoadConfigFile | 加载配置模板 |
| AVL\_SDK\_SetConfigInt | AVL SDK配置开关 |
| AVL\_SDK\_SetConfigString | AVL SDK配置开关 |
| AVL\_SDK\_GetConfigInt | AVL SDK获取当前配置 |
| AVL\_SDK\_GetConfigString | AVL SDK获取当前配置 |
| AVL\_SDK\_InitInstance | 根据配置信息初始化引擎实例 |
| AVL\_SDK\_Scan | 扫描函数 |
| AVL\_SDK\_QueryReportInt | 查询对象实例的检测结果 |
| AVL\_SDK\_QueryReportStr | 查询对象实例的检测结果 |
| AVL\_SDK\_Release | 释放引擎实例 |
| AVL\_SDK\_GetCurVersion | 获取当前引擎的版本 |
| AVL\_SDK\_GetLicenseExpDate | 获取当前引擎加载的License文件的日期 |

## 引擎结构定义

本节主要介绍开放引擎的结构体定义及相关数据宏定义

### 数据提供者（OBJ\_PROVIDER）结构体定义

对象提供者结构体主要用来用户与引擎交互数据，用户需要提供数据缓冲区以及缓冲区大小给引擎扫描，引擎也会通过这个结构体将产生的新数据传递给用户注册的回调函数。OBJ\_PROVIDER结构体的定义详情见engine.h文件。

* **obj\_ver:**

此结构体的版本。

* **obj\_type:**

数据类型（目前保留字段）。

* **evro\_type:**

引擎工作环境类型，详情见环境类型相关说明。

* **buf:**

数据缓冲区指针。

* **size:**

缓冲区中有效数据的大小。

* **obj\_des:**

数据描述信息。用户需要将数据的完整路径信息保存在此处。

### 数据处置者（OBJ\_DISPOSER）结构体定义

对象处置者结构体主要是用户用来向引擎传递处置回调函数用的。引擎在工作期间需要和用户交互（例如通知引擎中断扫描、获取扫描结果等），所以通过这个结构体传递交互用的接口函数。OBJ\_DISPOSER结构体的定义详情见engine.h文件。

* **rpt\_callback:**

用来接收引擎发给用户检测结果报告的回调函数指针，定义详见P\_OBJ\_DIPOSER\_CALLBACK。

* **p\_rpt\_param:**

用户传递给rpt\_callback回调函数的参数指针。

* **query\_continue\_callback:**

用来答复引擎是否中断扫描的回调函数指针，定义详见P\_QUERY\_CONTINUE\_CALLBACK。

* **p\_qc\_param:**

用户传递给query\_continue\_callback回调函数的参数指针。

## 开放引擎读写配置所用到的数据定义

开放引擎配置选项基于使用类型进行如下分类

### 数据报告（AVL\_SDK\_QueryReport函数)所用到的数据定义）

|  |  |
| --- | --- |
| 定义 | 说明 |
| RPT\_IDX\_OBJ\_DESCRIPTION | 数据对象的描述 |
| RPT\_IDX\_MALWARE\_ID | 获取引擎判断的恶意代码ID |
| RPT\_IDX\_ANALYSER | 获取报告MALWARE\_ID的检测模块的模块名 |
| RPT\_IDX\_CLOUD\_MALWARENAME\_ID | 获取云检测检出病毒名 |

### 开放引擎工作环境配置

表 2-2 开放引擎工作环境读写配置项

|  |  |  |  |
| --- | --- | --- | --- |
| 变量 | 类型 | 取值范围 | 说明 |
| **CFG\_STR\_MODULE\_PATH** | 字符串 | 根据填充 | 用来声明引擎子模块所在路径，可以将不同版本的模块放在不同路径，以方便不同的引擎实例去加载 |
| **CFG\_STR\_DATA\_PATH** | 字符串 | 根据填充 | 用来声明开放引擎特征库所在路径，可以将不同版本的引擎特征库放在不同路径，以方便不同的引擎实例去加载，从而实现引擎更新库热重启 |
| **CFG\_STR\_LICENSE\_PATH** | 字符串 | 根据填充 | 用来声明SDK引擎待使用License文件的完整路径，而不是License文件所在目录的路径 |
| **CFG\_STR\_LOG\_PATH** | 字符串 | 根据填充 | 用来声明引擎待使用记录日志目录路径 |

### 开放引擎模块配置

使用AVL\_SDK\_SetConfigInt/AVL\_SDK\_SetConfigString函数接口配置模块，如不设置则使用默认值。

表 2-3 开放引擎模块读写配置项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量 | 类型 | 取值范围 | 说明 | 默认值 |
| **CFG\_FLAG\_APACK\_ENABLE** | 布尔型 | 0或1 | 控制解包模块是否工作，此配置选项需要在开放引擎初始化实例之前设置，初始化之后调整无效。  0表示不加载，1表示加载。 | 1 |
| **CFG\_INT\_APACK\_MAX\_ARCHIVE\_SIZE** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置可拆解压缩包的大小，由于解压数据的多少会影响性能以及检出率，所以需要调整此配置选项以求平衡。这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。（单位：byte） | 100 \* 1024 \* 1024 |
| **CFG\_INT\_APACK\_MAX\_FILE\_NUM** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置可解压释放的文件数，由于解压数据的多少会影响性能以及检出率，所以需要调整此配置选项以求平衡。这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。 | 10000 |
| **CFG\_INT\_APACK\_MAX\_FILE\_SIZE** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置可解压释放的最大文件尺寸，由于解压数据的多少会影响性能以及检出率，所以需要调整此配置选项以求平衡。这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。（单位：byte） | 100 \* 1024 \* 1024 |
| **CFG\_INT\_APACK\_RECURE\_LAYER** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置可递归解压压缩包的次数，由于压缩包中可能会包含压缩包，而解包层数的多少会影响性能以及检出率，所以需要调整此配置选项以求平衡。这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。 | 5 |
| **CFG\_INT\_APACK\_MAX\_COMPR\_SCALE** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置可解压缩包的最大压缩比，由于压缩包中有可能会包含压缩比极大的数据，如果对其进行解压缩会极其占用系统资源，进而会影响性能、能耗以及检出率，所以需要调整此配置选项以求平衡。这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。 | 200 |
| **CFG\_INT\_APACK\_TIMEOUT** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置模块工作时间，由于解压操作是一件耗时的操作，所以需要调整此配置选项以求平衡。这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。（单位：s） | 5\*60 |
| **CFG\_INT\_APACK\_USE\_MAX\_MEMORY\_SIZE** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置模块工作时可以使用的内存的大小，如果解压出来的数据大小超过了限制，则后面的数据将会被丢弃，所以需要调整此配置选项以求平衡。置选项以求平衡。这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。 | 100 \* 1024 \* 1024 |
| **CFG\_FLAG\_APACK\_RPTBOMB** | 整数型 | 0 ~ 0xffffffff | 设置模块解包时是否反馈发现的包裹炸弹，这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。 | 0 |
| **CFG\_FLAG\_CLOUD\_DETECT\_ENABLE** | 布尔型 | 0或1 | 控制云检测模块是否工作，此配置选项需要在开放引擎初始化实例之前设置，初始化之后调整无效。0表示不加载，1表示加载。 | 1 |
| **CFG\_INT\_CLOUD\_CONNECT\_TIMEOUT** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置引擎连接云服务器的超时时间，初始化之后调整无效。 | 3 |
| **CFG\_INT\_CLOUD\_DETECT\_TIMEOUT** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置引擎云检测模块检测单个文件的超时时间，初始化之后调整无效。（单位：s） | 5 |
| **CFG\_INT\_CLOUD\_MAX\_SCAN\_FILE\_SIZE** | 整数型 | 0 ~ 0xffffffff | 本配置选项主要设置可检测的最大文件尺寸，这个配置选项在引擎初始化实例之后依然可以在任意时刻修改其值，以达到控制需求。（单位：byte） | 1 \* 1024 \* 1024 |
| **CFG\_INT\_CLOUD\_CACHE\_MAX\_RECORD\_SIZE** | 整数型 | 0 ~ 0xffffffff | 将最近查询的MD5检测结果进行缓存，下次检测同MD5效率更高 | 10000 |
| **CFG\_INT\_CLOUD\_CACHE\_VALID\_HOURS** | 整数型 | 0 ~ 0xffffffff | 缓存保存时间（单位：h） | 24 |

## 开放引擎API返回值

开放引擎API的常见返回值包括2类：成功、异常。具体含义见表2-4至2-5所示。

表 2-4成功返回

|  |  |  |
| --- | --- | --- |
| 宏 | 值 | 说明 |
| ERR\_SUCCESS | 0 | 成功 |

表 2-5异常返回

|  |  |  |
| --- | --- | --- |
| 宏 | 值 | 说明 |
| ERR\_UNKNOWN | -1 | 未知错误 |
| ERR\_INCORRECT\_PARAM | -2 | 参数错误 |
| ERR\_NO\_ENOUGH\_MEM | -3 | 内存不足 |
| ERR\_BAD\_LICENSE | -4 | 设置License失败 |
| ERR\_BAD\_AUTH\_CODE | -5 | 授权码错误 |
| ERR\_BAD\_ENGINE\_VER | -6 | 开放引擎版本错误 |
| ERR\_BAD\_DATABASE | -7 | 数据错误 |
| ERR\_INIT\_COMPONENT\_FAILED | -8 | 初始化失败 |
| ERR\_MISS\_IMPORTANT\_MOD\_FILE | -9 | 缺少关键模块 |
| ERR\_MISS\_IMPORTANT\_DATA\_FILE | -10 | 缺少关键特征库 |
| ERR\_BAD\_WORKING\_PATH | -11 | 工作路径错误 |
| ERR\_LOAD\_CONFIG | -12 | 加载配置文件失败 |
| ERR\_SET\_CONFIG | -13 | 设置配置参数错误 |
| ERR\_GET\_CONFIG | -14 | 获取参数错误 |

# API参考

引擎调用接口的主要API包括初始化引擎、扫描、释放引擎等。

## AVL\_SDK\_CreateInstance

AVL SDK引擎创建新引擎实例函数，该函数在加载配置选项之前调用

***long AVL\_SDK\_CreateInstance（***

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

***）;***

参数：

ppEngine

新创建的开放引擎实例，该参数传入指针地址，传出为新创建的开放引擎实例指针，不可设置为NULL。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表。

说明：

每调用此接口一次都会创建一个新的开放引擎实例，而开放引擎实例在初始化之前是不能用于检测的。

## AVL\_SDK\_LoadConfigFile

AVL SDK 开放引擎设置数据接收者，该函数在扫描之前调用。

***long AVL\_SDK\_LoadConfigFile（***

***/\*[in]\*/ void \*pEngine，***

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

szFilename

配置模板全路径指针，不可设置为NULL。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

用此接口加载配置模板，可以加载引擎提供的相关配置模板。

## AVL\_SDK\_SetConfigInt

AVL SDK引擎配置函数。

***long AVL\_SDK\_SetConfigInt（***

***/\*[in]\*/ void \*pEngine，***

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

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

CfgIdx

开放引擎配置项，通过该参数指定需要配置的开关，该参数不可设置为NULL。

lValue

引擎配置开关值，针对不同的配置开关有对应的配置值。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

此接口用来配置整型数作为配置值的配置开关。

## AVL\_SDK\_SetConfigString

AVL SDK引擎配置函数

***long AVL\_SDK\_SetConfigString（***

***/\*[in]\*/ void \*pEngine，***

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

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

CfgIdx

开放引擎配置项，通过该参数指定需要配置的开关，该参数不可设置为NULL。

lValue

开放引擎配置开关值，针对不同的配置开关有对应的配置值

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

此接口用来配置字符串作为配置值的配置开关。

## AVL\_SDK\_GetConfigInt

AVL SDK开放引擎获取当前配置函数(整数型)。

***long AVL\_SDK\_GetConfigInt（***

***/\*[in]\*/ void \*pEngine，***

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

***/\*[out]\*/long \*pValue ，***

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

CfgIdx

开放引擎配置项，通过该参数指定需要配置的开关，该参数不可设置为NULL。可配置开关详见AVLSDK\_conf\_idx.h文件中的声明或(这里加入链接指向配置开关项)。

pValue

用来获取配置项对应的值

default\_val

p\_Value默认值，用于获取配置项值失败时设置默认值

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

此接口用来获取当前开放引擎实例整数型配置值。

## AVL\_SDK\_GetConfigString

AVL SDK开放引擎获取当前配置函数(字符型)。

***long AVL\_SDK\_GetConfigString（***

***/\*[in]\* void \*pEngine，***

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

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

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

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

CfgIdx

开放引擎配置项，通过该参数指定需要配置的开关，该参数不可设置为NULL。可配置开关详见AVLSDK\_conf\_idx.h文件中的声明或(这里加入链接指向配置开关项)。

Buf

用来存储获取到的字符串。

BufLen

声明参数Buf的可用空间。

Default\_Buf

设置获取不到的默认值，通常设置为NULL。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

此接口用来获取当前开放引擎实例字符型配置值。

## AVL\_SDK\_InitInstance

AVL SDK开放引擎根据配置信息初始化引擎实例，加载特征数据库、知识库以及检测模块，初始化完成之后，该开放引擎实例将处于可用状态，此接口调用应该在创建了引擎实例之后。

***long AVL\_SDK\_InitInstance（***

***/\*[in]\*/ void \*pEngine，***

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

pVerificationCode

引擎授权验证码。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

此接口真正开始初始化引擎实例，AVL\_SDK\_CreateInstance接口仅仅是分配内存空间。如果开放引擎实例初始化失败，此接口会自动调用释放开放引擎实例的接口。

## AVL\_SDK\_Scan

AVL SDK 开放引擎扫描函数，扫描待处理对象

***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***

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

pObj

待扫描数据(数据提供者)，需用户填充该结构体并将指针交给扫描函数。其结构定义详见[OBJ\_VECTOR\_PROVIDER](#_数据提供者(OBJ_VECTOR_PROVIDER)结构体定义)结构定义

P\_OBJ\_DISPOSER\_VECTOR\_CALLBACK

[检测结果处理回调函数](#_用户需要实现接口)，引擎通过调用此回调函数通知用户检测结果。

p\_param

检测结果处理回调函数参数。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

此接口需要在开放引擎实例初始化之后调用，两个结构体详情见相关定义。

## AVL\_SDK\_Release

释放AVL SDK 开放引擎实例，包括数据库及检测模块等。

***long AVL\_SDK\_Release（***

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

释放一个指定的引擎实例，但不影响其他正在工作中的开放引擎实例。

## AVL\_SDK\_QueryReportInt

AVL SDK 引擎检测结果报告查询接口(整数型)。

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

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

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

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

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

***);***

pEngine

引擎实例指针，不可设置为NULL。

pRptHanlde

引擎传递给用户的报告句柄指针，不可设置为NULL。

key

引擎支持的报告项，报告项详见AVLSDK\_rpt\_idx.h文件中的声明或([2.5.1](#_数据报告（AVL_SDK_QueryReport函数)所用到的数据定义))，不可设置为NULL。

value

用户提供的用来保存报告项对应结果的缓冲区指针，不可设置为NULL。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

该接口的引擎实例指针可以通过P\_OBJ\_DISPOSER\_CALLBACK回调函数的对应参数传递，报告项句柄指针同样在该回调函数传递给用户。

## AVL\_SDK\_ QueryReportString

AVL SDK 引擎检测结果报告查询接口(字符型)。

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

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

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

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

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

***);***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

pRptHanlde

引擎传递给用户的报告句柄指针，不可设置为NULL。

key

引擎支持的报告项，报告项详见AVLSDK\_rpt\_idx.h文件中的声明或([2.5.1](#_数据报告（AVL_SDK_QueryReport函数)所用到的数据定义))，不可设置为NULL。

value

用户提供的用来保存报告项对应结果的缓冲区指针的指针，不可设置为NULL。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

该接口的引擎实例指针可以通过P\_OBJ\_DISPOSER\_CALLBACK回调函数的对应参数传递，报告项句柄指针同样在该回调函数传递给用户。

## AVL\_SDK\_GetLicenseExpDate

获取当前开放引擎的License文件的日期。

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

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

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

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

***）;***

参数：

pEngine

开放引擎实例指针，不可设置为NULL。

buf

用来保存License时间的缓冲区，不可设置为NULL，且该缓冲区长度不能小于32字节。

len

用来声明该缓冲区的长度。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

如果缓冲区长度不足以保存引擎版本字符串，开放引擎会将该缓冲区填满但会保留最后一个字节为’\0’，同时函数不会返回异常，建议该缓冲区大小不小于128字节。

## AVL\_SDK\_ GetCurVersion

获取当前引擎的发行版本。

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

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

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

***);***

参数：

buf

用来保存引擎版本的缓冲区，不可设置为NULL，且该缓冲区长度不能小于32字节。

len

用来声明该缓冲区的长度。

返回结果：

返回错误码。具体值参见[2.6章开放引擎API返回值](#_引擎API返回值)列表

说明：

如果缓冲区长度不足以保存引擎版本字符串，引擎会将该缓冲区填满但会保留最后一个字节为’\0’，同时函数不会返回异常，建议该缓冲区大小不小于128字节。