

Protect SDK — Windows

配置为 JSON 字符串。必填 `access_key`；其余可选，不传则使用默认。

键	默认	说明
<code>access_key</code>	(必填)	接入密钥
<code>fast_tick_enabled</code>	<code>false</code>	按接入要求开启
<code>log_enabled</code>	<code>false</code>	<code>true</code> 时输出调试日志

示例：`{"access_key": "YOUR_ACCESS_KEY"}`

本目录按 `x86 / x86_64` 分开放，请与目标程序位数一致。

包内容

路径	说明
<code>lib/protect.dll</code>	SDK
<code>include/sdk_windows.h</code>	头文件
<code>demo/protect_demo.exe</code>	示例程序
<code>demo/main.cpp</code>	示例源码

API

函数	含义
<code>protect_start(config)</code>	启动。 <code>config</code> 为 JSON 字符串
<code>protect_stop()</code>	停止
<code>protect_error()</code>	最近一次错误，编码为系统 OEM 代码页（中文 Windows 一般为 GBK），不是 UTF-8。无错误为 ""。返回的指针在下次 <code>start / stop / error</code> 之前有效，需要保留请立刻拷贝

返回 `0` 表示成功，其它为错误码。失败后调用 `protect_error()`。

调用约定为 `__stdcall`。

在进程开始工作前 `start`，在退出或卸载库之前 `stop`：

程序类型	protect_start	protect_stop
控制台	初始化完成后	main 返回前；若处理 Ctrl+C，先 stop 再退出
带窗口的程序	窗口和资源就绪后	窗口销毁或消息循环结束、进程退出前
Windows 服务	开始处理请求前	收到停止控制时
自行加载 DLL	加载成功并取得函数地址后	卸载 DLL 之前

可以用 `atexit()` 作补充，但不能代替正常退出路径上的 `stop`。

Demo 在启动和停止之间等待 30 秒，只是留出一段运行时间。正式程序里换成自己的主逻辑，不必保留这段等待。

集成步骤

编译时链接

编译时加上头文件目录，运行时把 `protect.dll` 放在 `exe` 同目录或 `PATH` 中。

```
#include "sdk_windows.h"

int rc = protect_start("{\\"access_key\\":\\"YOUR_ACCESS_KEY\\"}");
if (rc != 0) {
    const char* err = protect_error();
}
protect_stop();
```

运行时加载

参考 `demo/main.cpp`: `LoadLibrary("protect.dll")` 后取函数地址。`stop` 必须在 `FreeLibrary` 之前调用。

运行 Demo

1. 把 `lib/protect.dll` 复制到 `demo/`（与 `protect_demo.exe` 同目录）。
2. 运行：

```
cd demo
protect_demo.exe YOUR_ACCESS_KEY
```

程序会先启动，等待约 30 秒，再停止并退出。

```
protect_demo.exe --help
```

若找不到 `protect.dll`，程序会提示放到 `protect_demo.exe` 所在目录。