Windows API Hook SDK Manual

Validtec Software, Inc.
http://www.validtec.com

API Hook SDK is a Software Development Kit for hooking Windows 32bit API functions, it will call your own function instead of Windows call some API. API Hook SDK also can hookthe functions in 3rd Appliction's DLL.

Table of contents

Features:

1. Your own functions have the same form as the API functions you need to hook, it is easy to use.
2. Hooks process in real time, that means it will install hook just when the process has created.
3. Takes very little system resource and nearly no CPU time consumed.

File List:

HookDemo.exe: Call HookSetup.dll to Setup API Hook
HookSetup.dll: Setup API Hook DLL
APIHookxp.dll: Win32 API Hook DLL for WINNT/2000/XP
APIHook9x.dll: Win32 API Hook DLL for WIN9X/WINME
HookFunc.dll: Your own DLL include functions that will be called when hook occurred

APIHook Directory: Source code of Win32 API Hook DLL
ADFilter Directory: Source code of Internet Advertising Filter example
DESCrypt Directory: Source code of DES Cryptological Library
FileDelphi Directory: Source code of File access hook function Delphi example
FileHook Directory: Source code of File access hook function example
HookDemo Directory: Source code of HookDemo program
HookSetup Directory: Source code of API Hook setup DLL
MsgHook Directory: Source code of system message hook function example
NetCrypt Directory: Source code of network transmission crypto function example
ProcHide Directory: Source code of hide process from taskmanager function example

RegHook Directory: Source code of registry hook function example
SocketDelphi Directory: Source code of socket hook function Delphi example
SocketHook Directory: Source code of socket hook function example

Usage:

Developer only need write the function related with the functions you want to hook, compiled HookFunc.dll, and call install and uninstall function at HookSetup.dll, it will implement the hook.

HookFunc project:

1. Create DLL project file - HookFunc
2. If the project name is not HookFunc, modify the output file name to HookFunc.dll
3. Modify HookFunc.cpp and HookFunc.def
4. Compile and link the project, it will output file HookFunc.dll

HookFunc project contain something as below:

1) HookFunc.h, there is only CAPIINFO structure definite in this file:

```
#ifndef _HookFunc_h_
#define _HookFunc_h_

typedef struct
{
char *module_name;
char *func_name;
char *c_func_name;
}CAPIINFO;
```

module_name is the name of a DLL or other module file name that will be hooked, for example : kernel32.dll

func_name is the function name and parameters of user's DLL that will be hooked, like C style format ,for examples:
connect(SOCKET, struct sockaddr *, INT)

c_func_name is your own function that called as the related function hooked, for examples:
cConnect(SOCKET s, struct sockaddr *name, int namelen).

2) Define CAPIINFO c_api_info[] and fill it, must be NULL in the end, It is the hook function information, for examples:

```
CAPIINFO c_api_info[] = {
{"WSOCK32.DLL", "socket(INT, INT, INT)", "cSocket"},
{"WSOCK32.DLL", "connect(SOCKET, struct sockaddr *, INT)",
"cConnect"},
{"WSOCK32.DLL", "recv(INT, char *, INT, INT)", "cRecv"},
{"WSOCK32.DLL", "send(INT, char *, INT, INT)", "cSend"},
{"ADVAPI32.DLL", "RegOpenKeyA(HKEY, LPCSTR, PHKEY)",
```

```
"cRegOpenKeyA"},
{NULL,NULL,NULL} }; //must contain this
```

3) Must be define function GetCAPIINFO:

```
CAPIINFO *GetCAPIINFO()
{ return &c_api_info[0]; }
```

4) Coding with user function, for example:

```
DWORD _cdecl cFuncName(type1 param1, type2, param2, …)
```

The function must be defined with WINAPI (in Delphi it is stdcall), the function definite must be as same as original function, for example:

```
int WINAPI cConnect(SOCKET s, struct sockaddr *name, int
namelen)
{
struct sockaddr_in *paddr =(struct sockaddr_in *)name;
char *ip =inet_ntoa(paddr->sin_addr);
int port =ntohs(paddr->sin_port);
int ret =connect(s, name, namelen);
int err=WSAGetLastError();
WriteLog("connect: ip=%s, port=%d, ret=%d\n", ip, port,
ret); // check filter
WSASetLastError(err);
return ret; }
```

in the example, we recovery error code, because when we handle with our own procedure, the error code will change, and the original process could be do next according to the error code.even cConnect not call original connect, we must use WSASetLastError or SetLastError to set error code when cConnect return. Other functions should do as the same.

By the way, in win9x, it is best to call the original API functions in HookFunc for HookFunc will call the original dll module, otherwise before call HookFunc, first use LoadLibrary() to load original dll file in apihook application, hook could be failed if not do like this sometimes.

5) HookFunc.def contains GetCAPIINFO and users functions exports, for example:

```
LIBRARY HookFunc

EXPORTS
    GetCAPIINFO            @1
    cRegOpenKeyA           @2
    cRegOpenKeyW           @3
    cRegQueryValueA        @4
    cRegQueryValueW        @5
    cRegQueryValueExA      @6
    cRegQueryValueExW      @7
```

HookSetup.dll

HookSetup.dll contain these functions (refer to HookSetup.h):

1) int WINAPI InitAPIHook();
2) int WINAPI HookAllProcesses();
3) int WINAPI UnhookAllProcesses();
4) int WINAPI NTHookProcess(DWORD process_id);
5) int WINAPI NTUnhookProcess(DWORD process_id);
6) int WINAPI NTHookProcess2(char *mod_name);
7) int WINAPI NTUnhookProcess2(char *mod_name).

Compile and run

Build the APIHook,HookSetup and HookFunc projects, put APIHookxp.dll (or APIHook9x.dll),HookSetup.dll,HookFunc.dll and your application which use the hook functions (e.g. Hookdemo.exe) to the same directory,then run your application, that's done.

Validtec Software, Inc.
http://www.validtec.com
January 28, 2004