Forcible Insertion

Making your programs transcend the limits

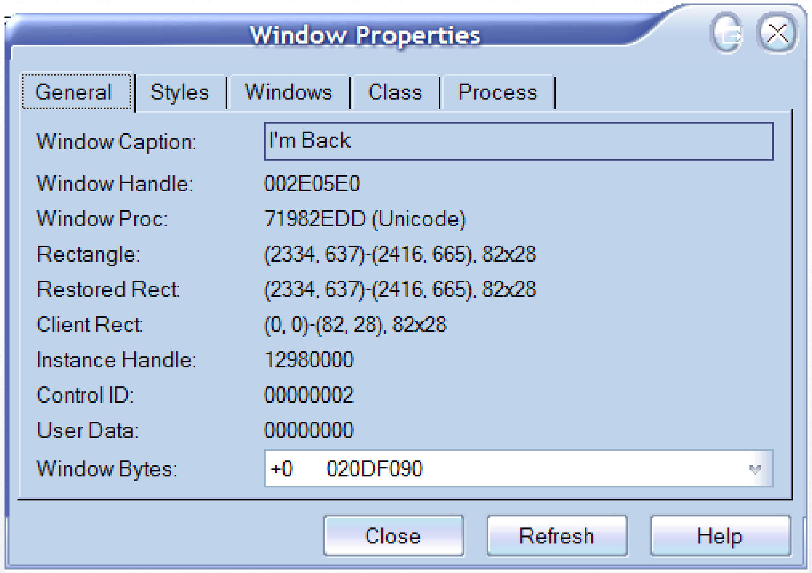
"or breaking windows"
By Steve Hanna

Window Handles

- Every item in the windows GUI has a handle
- All GUI objects are windows with their own handles
- Relationship between objects, Parent and Child windows
- GUI Object types are called Classes.
- Button, Edit, Static, Listbox, ComboBox, are all examples of GUI Objects

Obtaining Window Handles

- Programmatically
  - FindWindow, FindWindowEx
  - EnumWindows
- On the Fly
  - Spy++ (Visual C++ tool)
  - Example: AIM "Back" button



Window Caption:	I'm Back
Window Handle:	002E05E0
Window Proc:	71982EDD (Unicode)
Rectangle:	(2334, 637)-(2416, 665), 82x28
Restored Rect:	(2334, 637)-(2416, 665), 82x28
Client Rect:	(0, 0)-(82, 28), 82x28
Instance Handle:	12980000
Control ID:	00000002
User Data:	00000000
Window Bytes:	+0 020DF090

Examples

- FindWindow and FindWindowEX return HWND, the data type for windows handles
- HWND FindWindow(LPCTSTR lpClassName, LPCTSTR lpWindowName);
- HWND m_hWnd = FindWindow(NULL, ”WindowNameHere”);
- Window Class or Window Title can be blank, but not both
- EnumWindows – We’ll look at later

Windows Message System

- What are Messages?
    - Messages are pieces of information sent from one program or another encouraging or notifying it that it should take some action
- When are messages generated?
    - Messages are generated when any event takes place in the system
    - Mouse movement, redraw a window, hit a key, etc. We'll look at the messages later

Message Details

- Two types of messages, System and Application defined messages
- Two types of message routing, Queued and nonqueued.

Sending Messages

- LRESULT SendMessage (HWND hWnd, UINT Msg, WPARAM wParam,LPARAM lParam);
- Example
  - LRESULT lrResult = SendMessage(myHWnd, WM_CLOSE, NULL, NULL)
- Example
  - wParam and lParam are used to specify arguments to the messages. As show above WM_CLOSE doesn't require additional requirements.

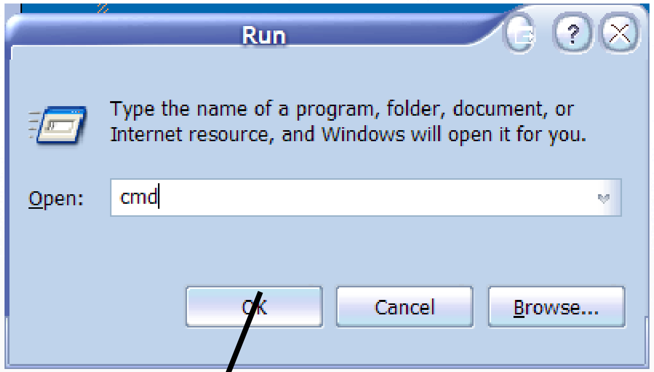
Dealing with Messages

- The Windows Message Loop

```
while( (bRet = GetMessage( &msg, NULL, 0, 0 )) != 0)
  {
     if (bRet == -1)
     {
        // handle the error and possibly exit
     }
     else
     {
         switch(msg.message)
         {
             case WM_RBUTTONDOWN:
                     //dostuff
             break;
           //process other messages
         }

        TranslateMessage(&msg);
        DispatchMessage(&msg);
     }
  }
```

Example

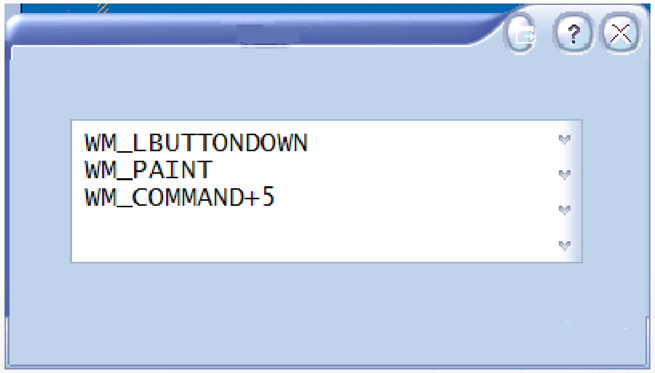


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

WM_LBUTTONDOWN

"Message Queue"



```
WM_LBUTTONDOWN
WM_PAINT
WM_COMMAND+5
```

Program calls function that corresponds to the event that took place

Hooking

- What is hooking?
  - Hooking is injecting code to handle other window events before the window has a chance to process them
- Uses?
  - Key logging, Sub classing (Out of Process), Plugins

The Hook Chain

- If a hook is installed it is put into the system wide "Hook Chain"
- Example \\ signifies the message path

System
\\
Keylogger
\\
Application

The hook chain can very long, just remember there is a performance cost for using them and too many may make the user become irritated or suspicious.

Hook API

- SetWindowsHookEx
- Types of Hooks – see MSDN for more information
  - WH_CALLWNDPROC
  - WH_CALLWNDPROCRET
  - WH_CBT
  - WH_DEBUG
  - WH_FOREGROUNDIDLE
  - WH_GETMESSAGE
  - WH_JOURNALPLAYBACK
  - WH_JOURNALRECORD hook procedure. For more information, see the WH_JOURNALRECORD
  - WH_KEYBOARD
  - WH_KEYBOARD_LL
  - WH_MOUSE
  - WH_MOUSE_LL
  - WH_MSGFILTER
  - WH_SHELL
  - WH_SYSMSGFILTER

```
HHOOK SetWindowsHookEx(int idHook, HOOKPROC lpfn,
    HINSTANCE hMod, DWORD dwThreadId );
```

Example Hook

- HHOOK msg=SetWindowsHookEx(WH_GETMESSAGE, (HOOKPROC)GetMsgProc,hins,0);
- First Parameter: Hook type
- Second Parameter: Callback Function
- Third Parameter: VERY IMPORTANT, specifies the ThreadID which the hook will be associated with. In order to be associated with all windows this parameter must be ZERO!
- Real example code later.

Example Keylogger

- Basic program that intercepts all keyboard messages and writes them to a specified file.
- Also sets attributes so file is harder to find.
- Let’s view the example!

Security

- Messages don’t have access privileges.
- Any messages can be sent or intercepted from any window.