Practical Malware Analysis

Ch 5: IDA Pro

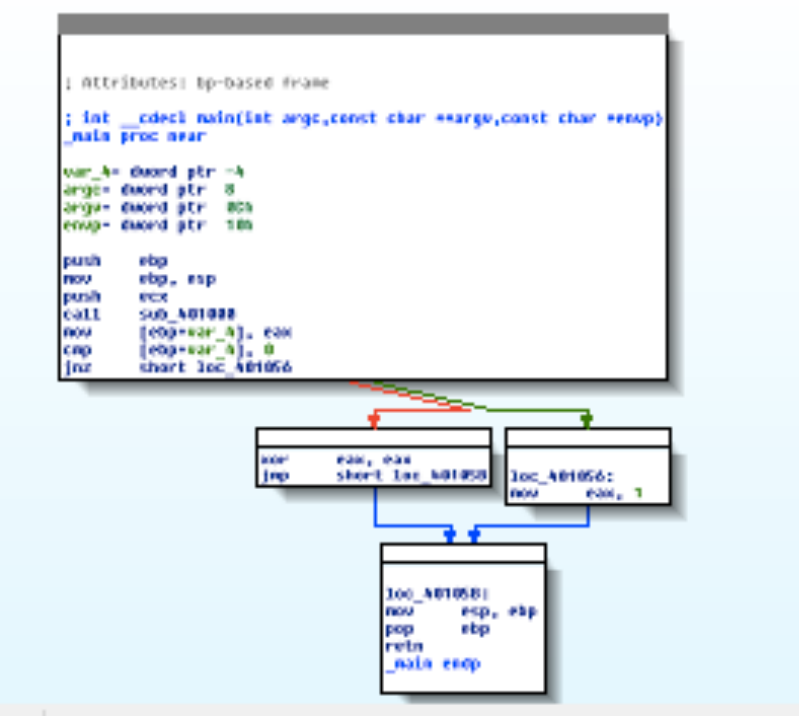
Last modified
9-28-21

IDA Pro Versions

- Full-featured pay version
- Old free version
  - Both support x86
  - Pay version supports x64 and other processors, such as cell phone processors
- Both have code signatures for common library code in FLIRT (Fast Library identification and Recognition Technology)

Graph and Text Mode

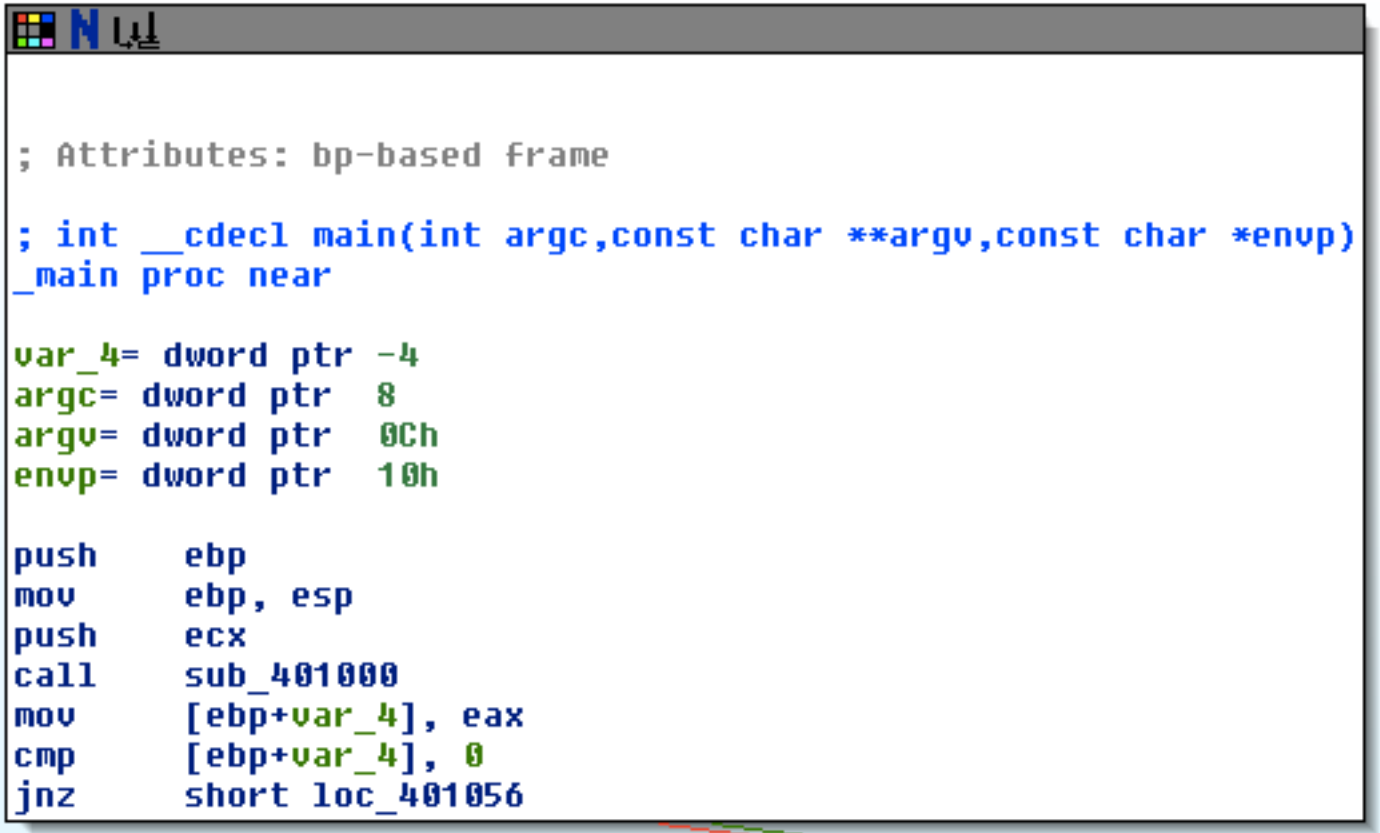
- Spacebar switches mode



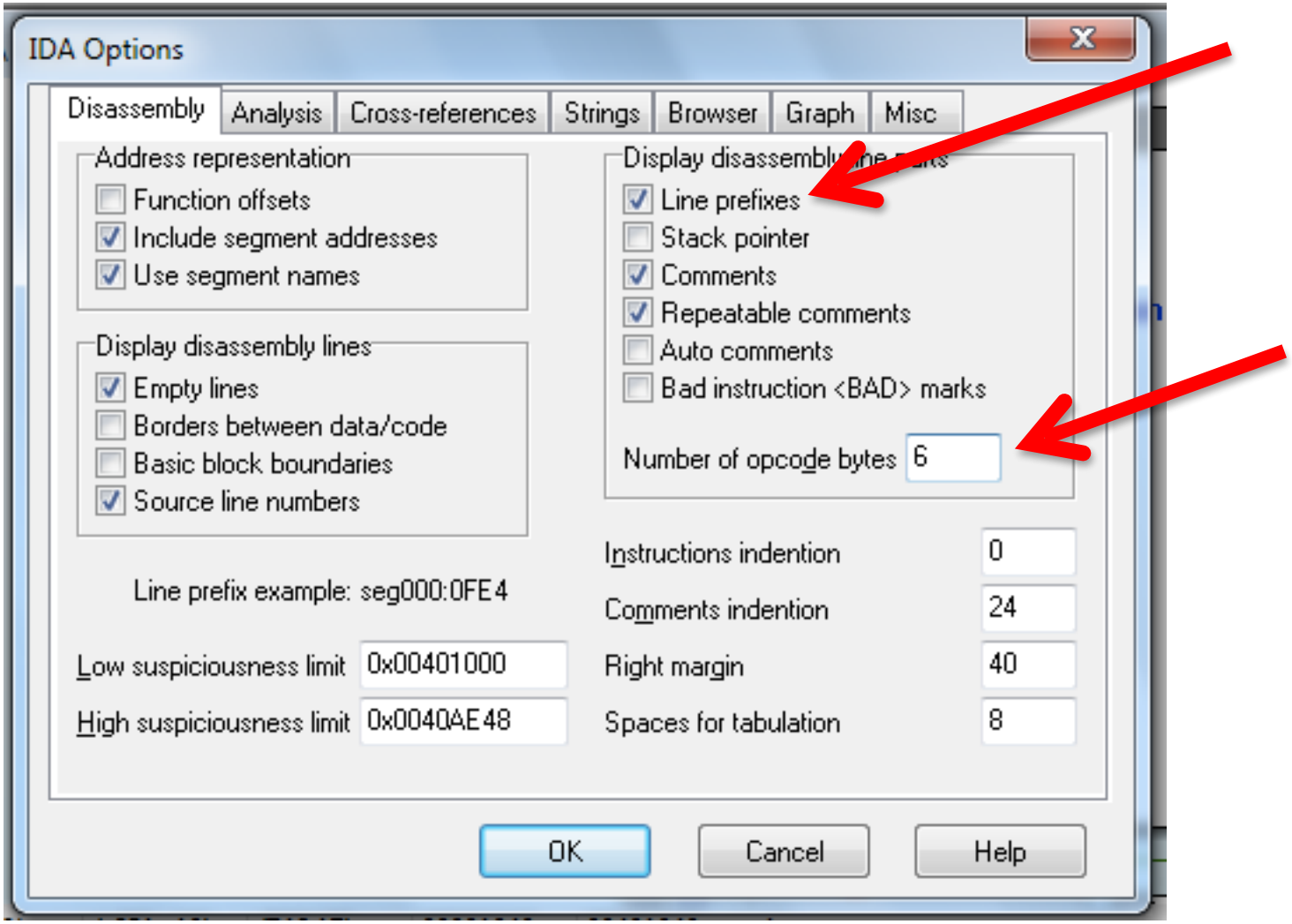
IDA View-A

```
.text:00401040
.text:00401040 ; Attributes: bp-based frame
.text:00401040
.text:00401040 ; int __cdecl main(int argc,const char **argv,const char *envp)
.text:00401040 _main            proc near               ; CODE XREF: start+AF↓p
.text:00401040
.text:00401040 var_4            = dword ptr -4
.text:00401040 argc             = dword ptr  8
.text:00401040 argv             = dword ptr  0Ch
.text:00401040 envp             = dword ptr  10h
.text:00401040
.text:00401040                  push    ebp
.text:00401041                  mov     ebp, esp
.text:00401043                  push    ecx
```

Default Graph Mode Display



Options, General



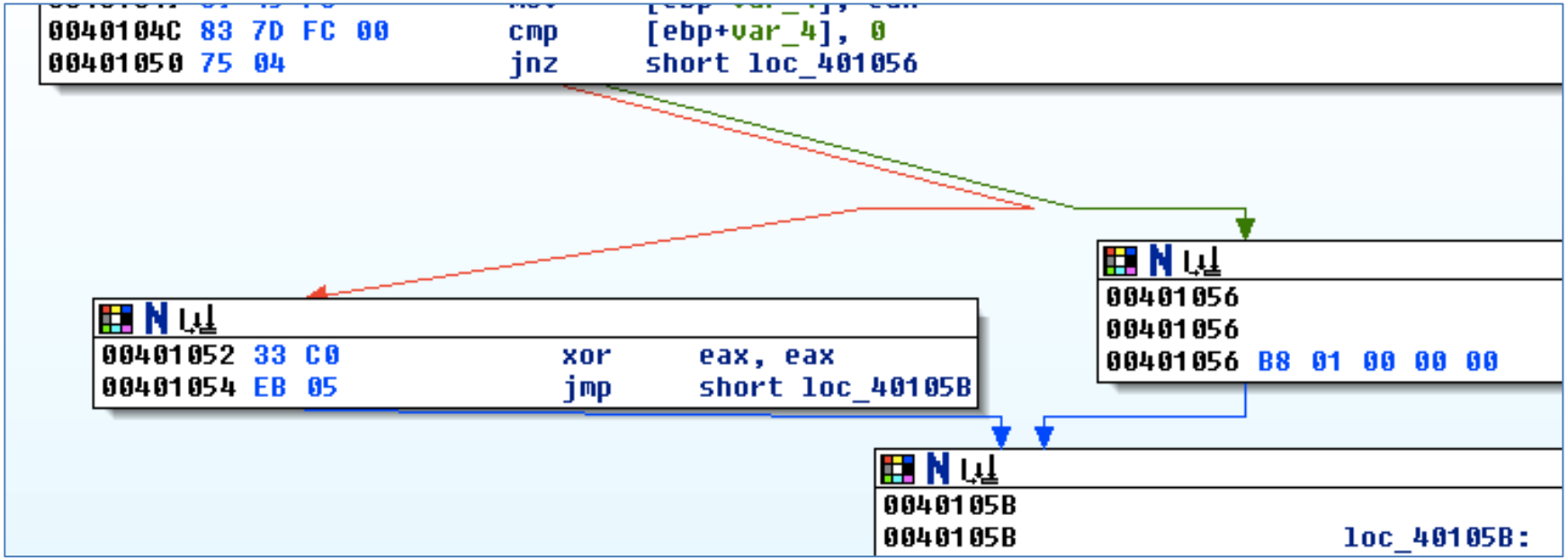
Better Graph Mode View

```
00401040
00401040
00401040                        ; Attributes: bp-based frame
00401040
00401040                        ; int __cdecl main(int argc,const char **argv,const char *envp)
00401040                        _main proc near
00401040
00401040                        var_4= dword ptr -4
00401040                        argc= dword ptr  8
00401040                        argv= dword ptr  0Ch
00401040                        envp= dword ptr  10h
00401040
00401040 55                     push    ebp
00401041 8B EC                  mov     ebp, esp
00401043 51                     push    ecx
00401044 E8 B7 FF FF FF         call    sub_401000
00401049 89 45 FC               mov     [ebp+var_4], eax
0040104C 83 7D FC 00            cmp     [ebp+var_4], 0
00401050 75 04                  jnz     short loc_401056
```

Arrows

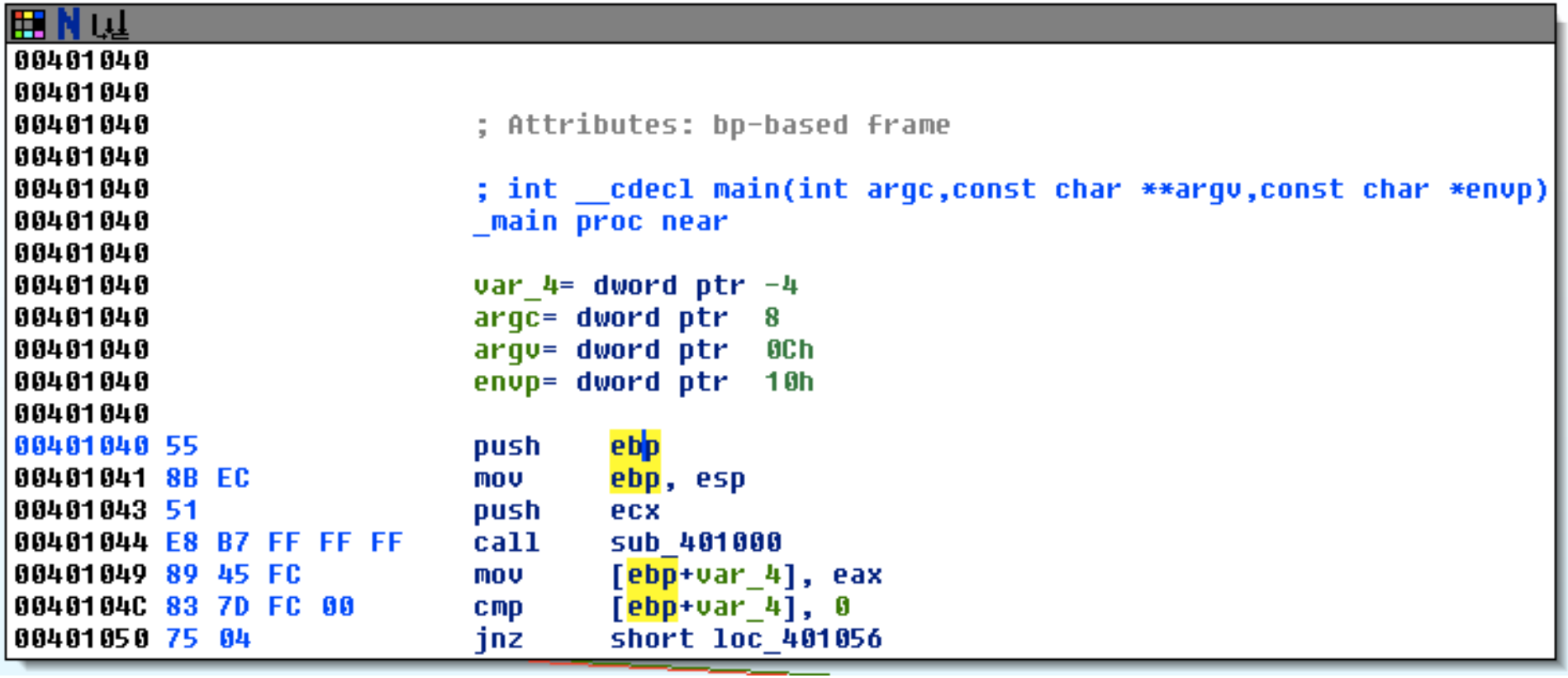
- Colors
  - Red — Conditional jump not taken
  - Green — Conditional jump taken
  - Blue — Unconditional jump
- Direction
  - Up — Loop

Arrow Color Example



Highlighting

- Highlighting text in graph mode highlights every instance of that text



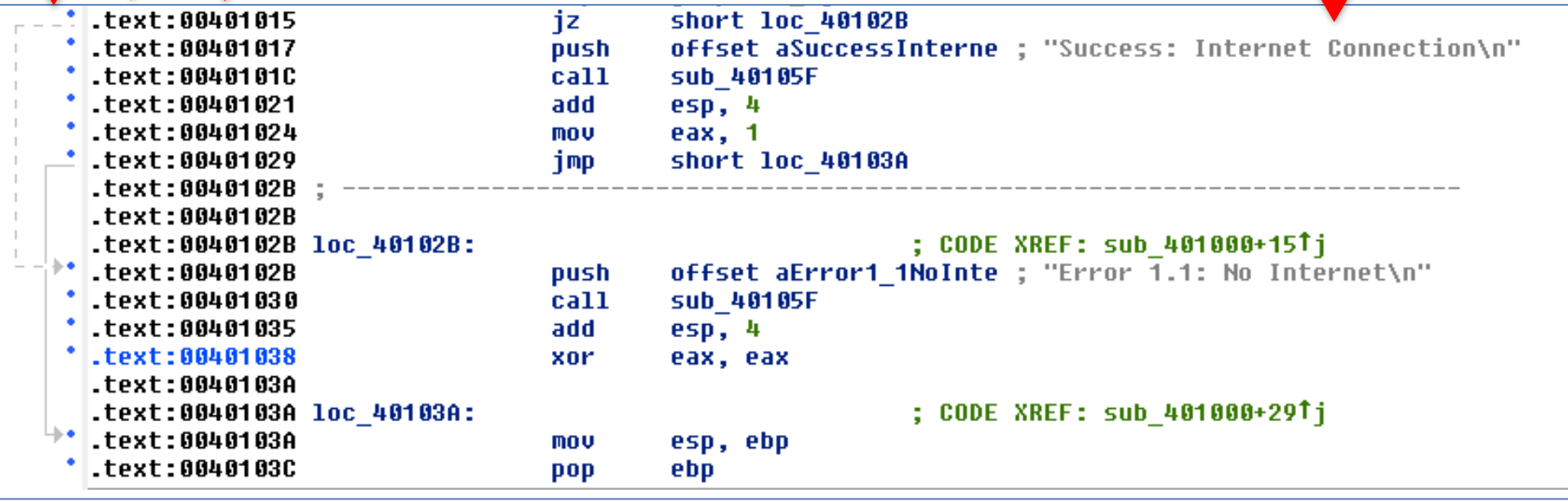
Text Mode

Arrows
Solid = Unconditional
Dashed = Conditional
Up = Loop

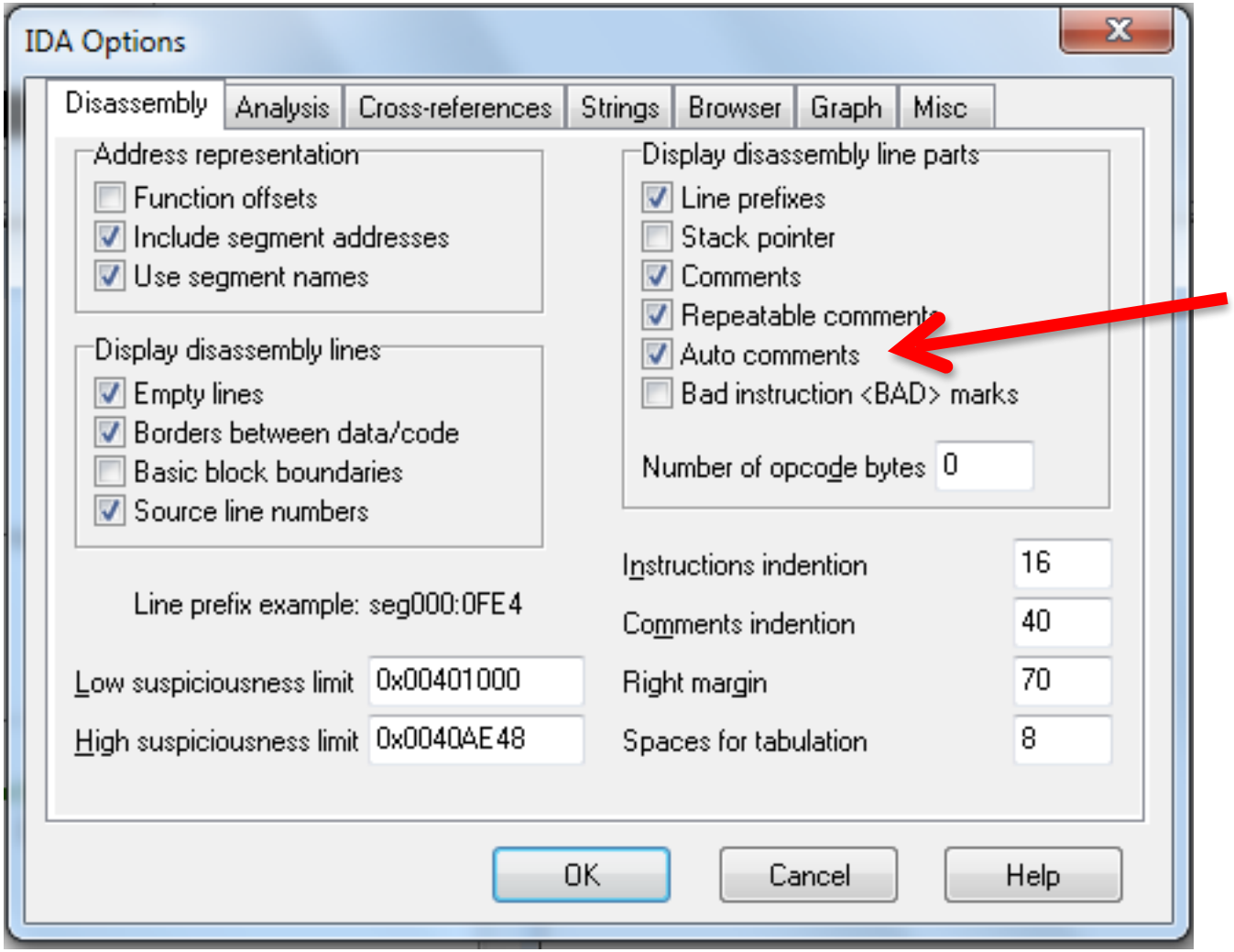
Section

Address

Comment Generated by IDA Pro



Options, General



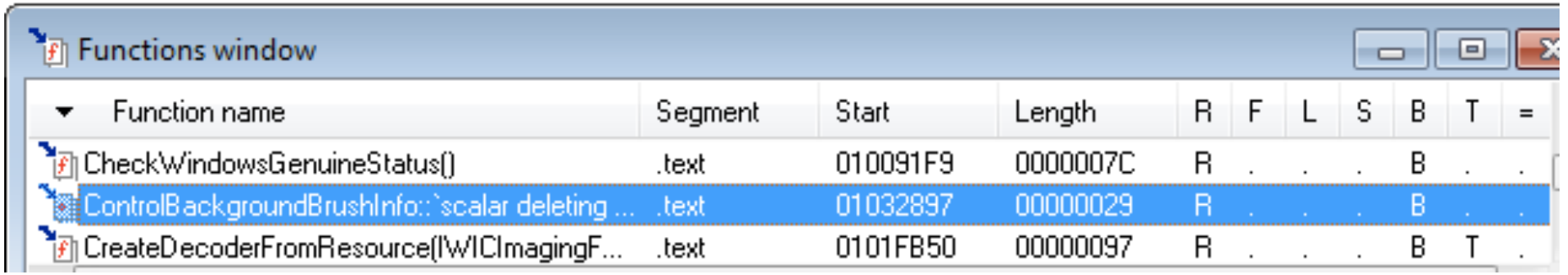
Adds Comments to Each Instruction

```
.text:00401015                  jz      short loc_40102B ; Jump if Zero (ZF=1)
.text:00401017                  push    offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C                  call    sub_40105F      ; Call Procedure
.text:00401021                  add     esp, 4          ; Add
.text:00401024                  mov     eax, 1
.text:00401029                  jmp     short loc_40103A ; Jump
.text:0040102B ; ---------------------------------------------------------------------------
.text:0040102B
.text:0040102B loc_40102B:                              ; CODE XREF: sub_401000+15↑j
.text:0040102B                  push    offset aError1_1NoInte ; "Error 1.1: No Internet\n"
.text:00401030                  call    sub_40105F      ; Call Procedure
.text:00401035                  add     esp, 4          ; Add
.text:00401038                  xor     eax, eax        ; Logical Exclusive OR
.text:0040103A
.text:0040103A loc_40103A:                              ; CODE XREF: sub_401000+29↑j
.text:0040103A                  mov     esp, ebp
.text:0040103C                  pop     ebp
```

Useful Windows for Analysis

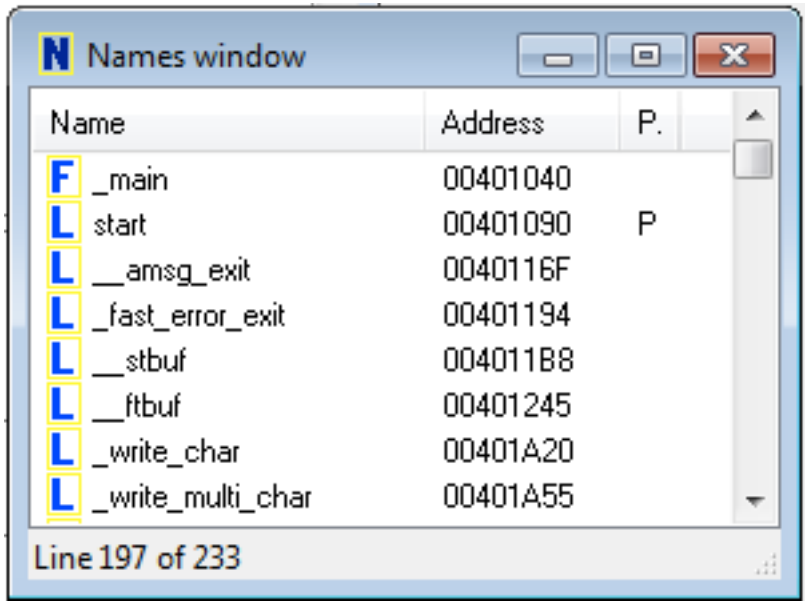
Functions

- Shows each function, length, and flags
  - L = Library functions
- Sortable
  - Large functions usually more important

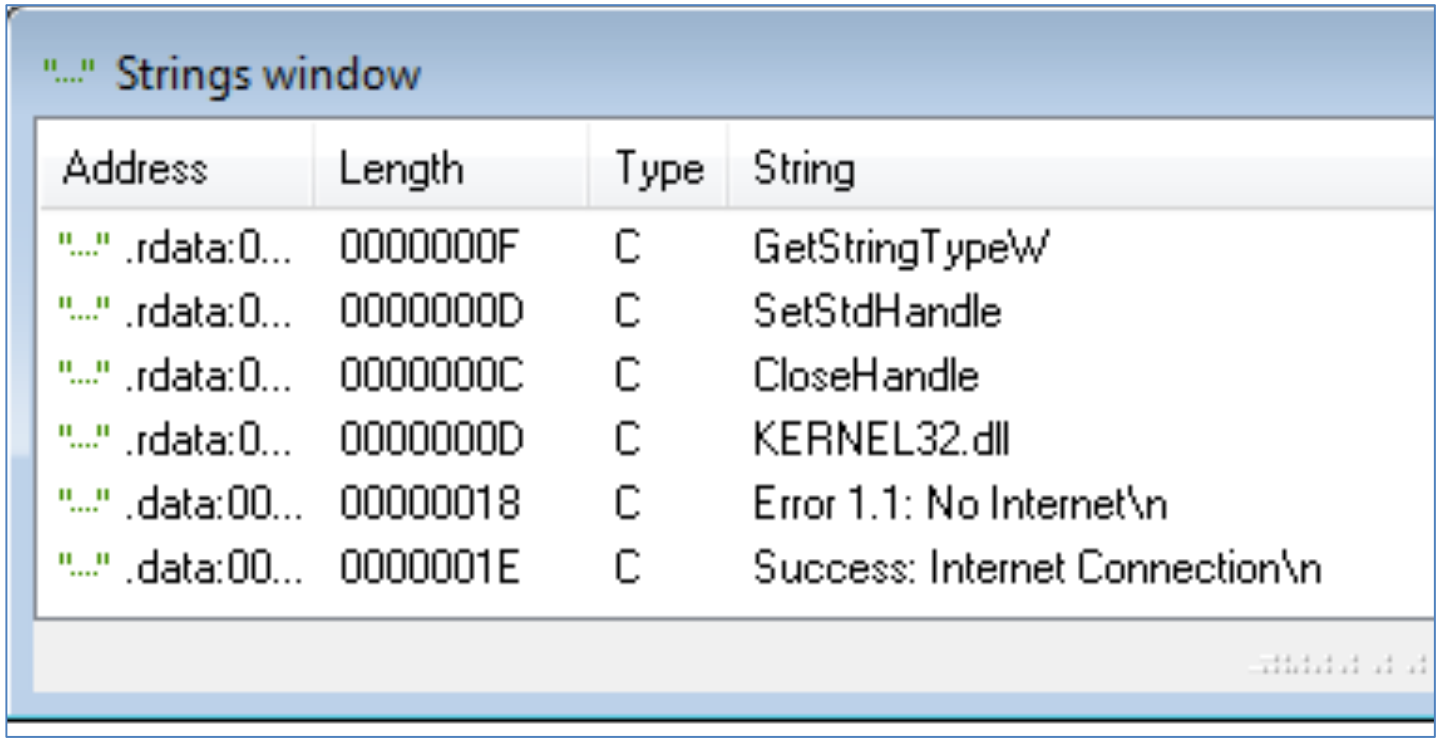


Names Window

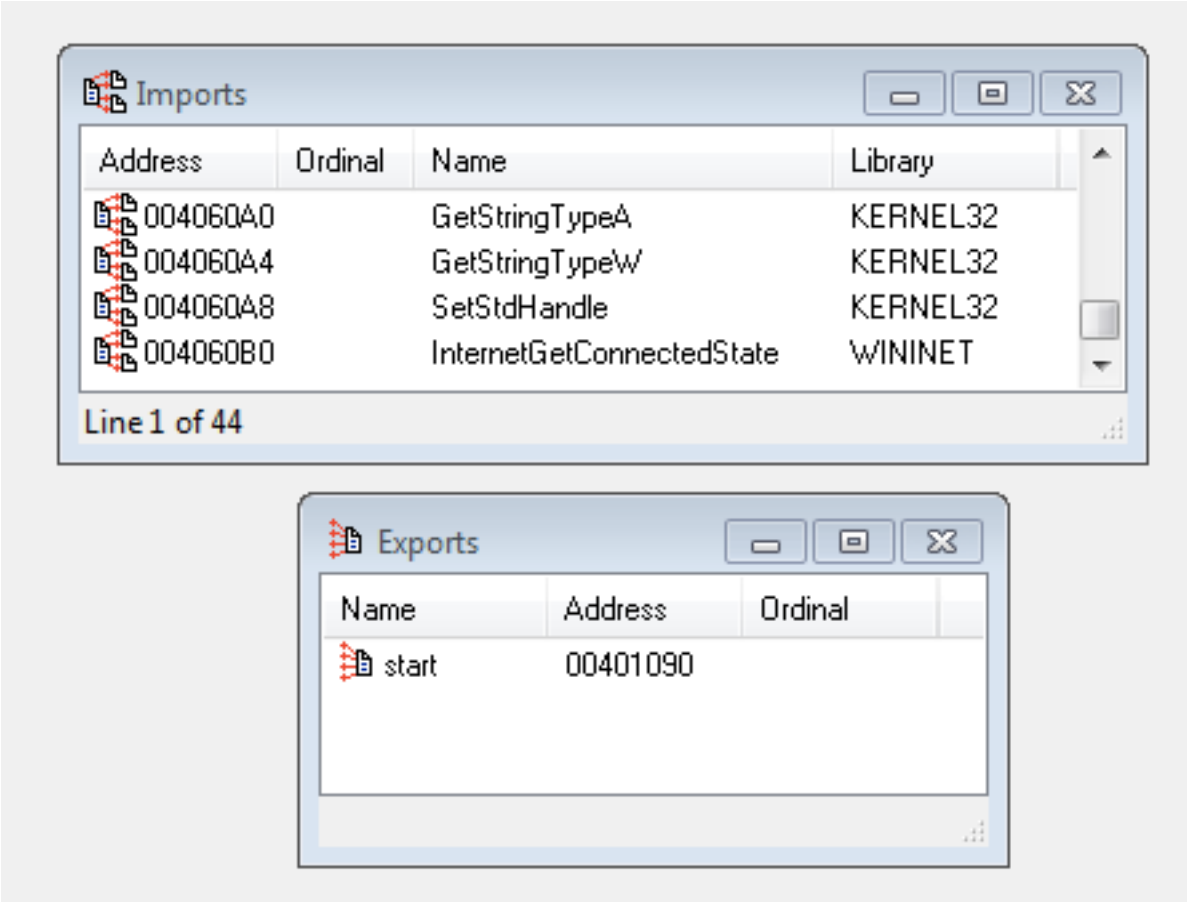
- Every address with a name
  - Functions, named code, named data, strings



Strings

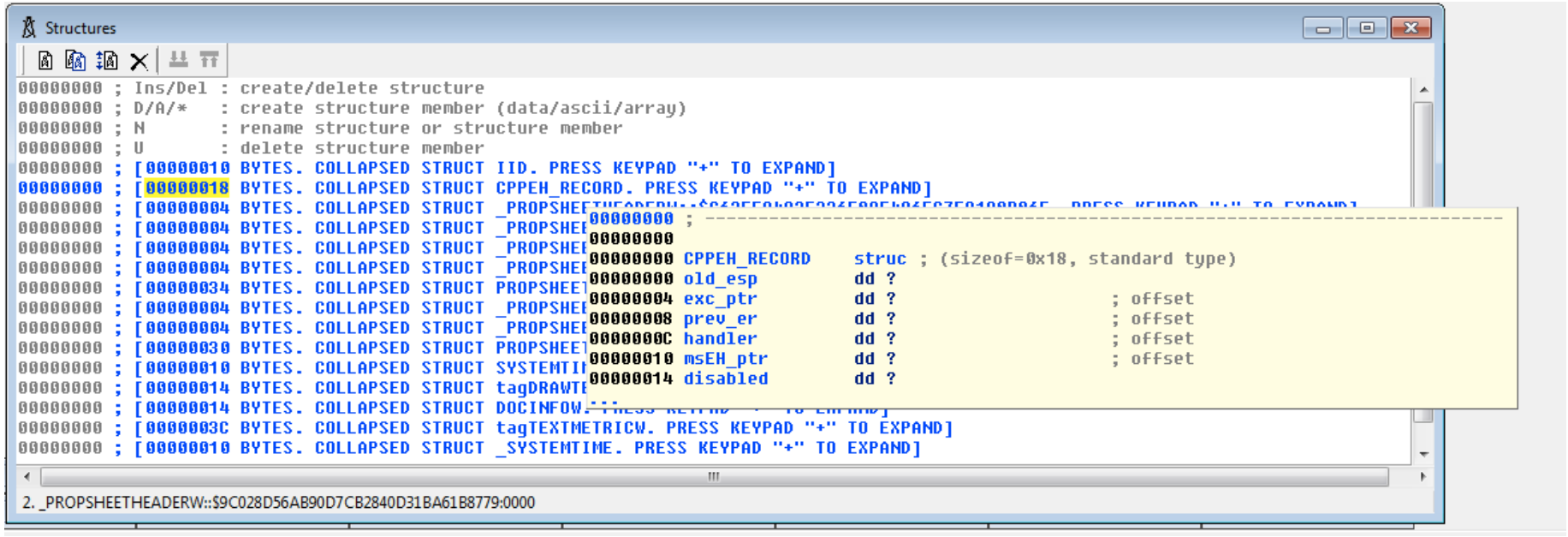

Strings window
Address
Length
Type
String
.rdata:0...
0000000F
C
GetStringTypeW
.rdata:0...
0000000D
C
SetStdHandle
.rdata:0...
0000000C
C
CloseHandle
.rdata:0...
0000000D
C
KERNEL32.dll
.data:00...
00000018
C
Error 1.1: No Internet\n
.data:00...
0000001E
C
Success: Internet Connection\n

Imports & Exports



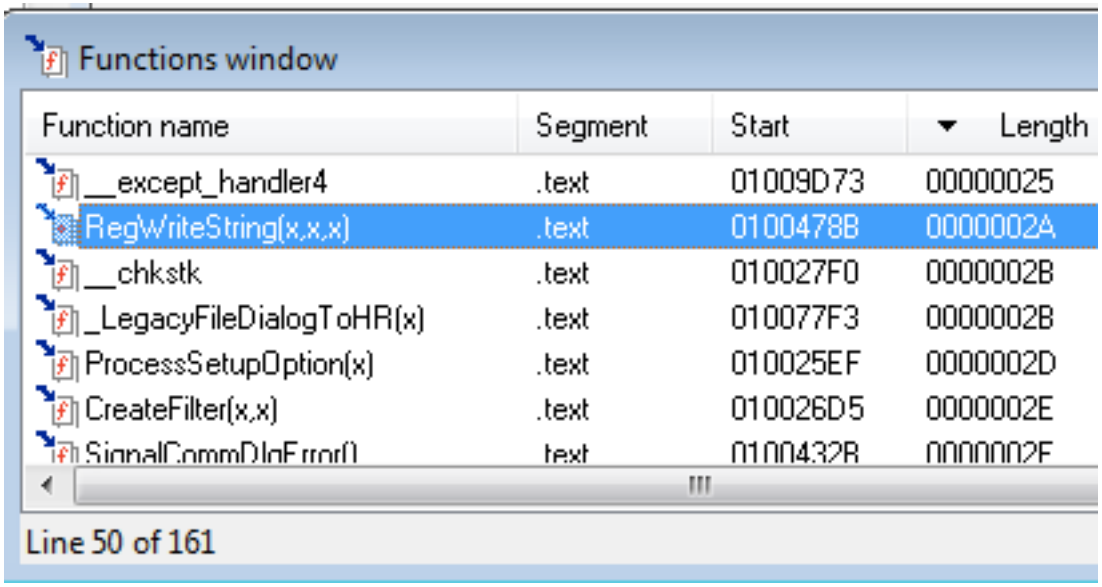
Structures

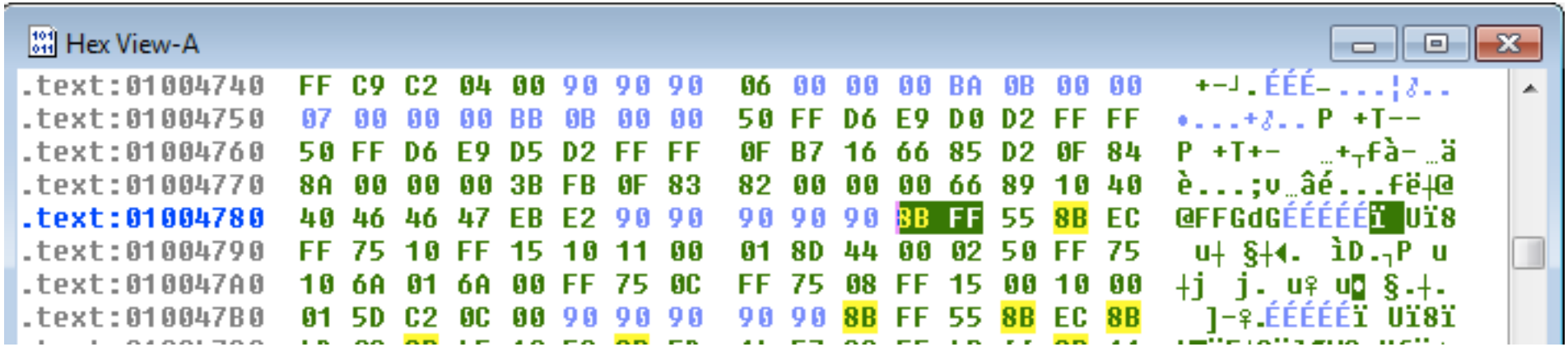
- All active data structures
  - Hover to see yellow pop-up window



Cross-Reference

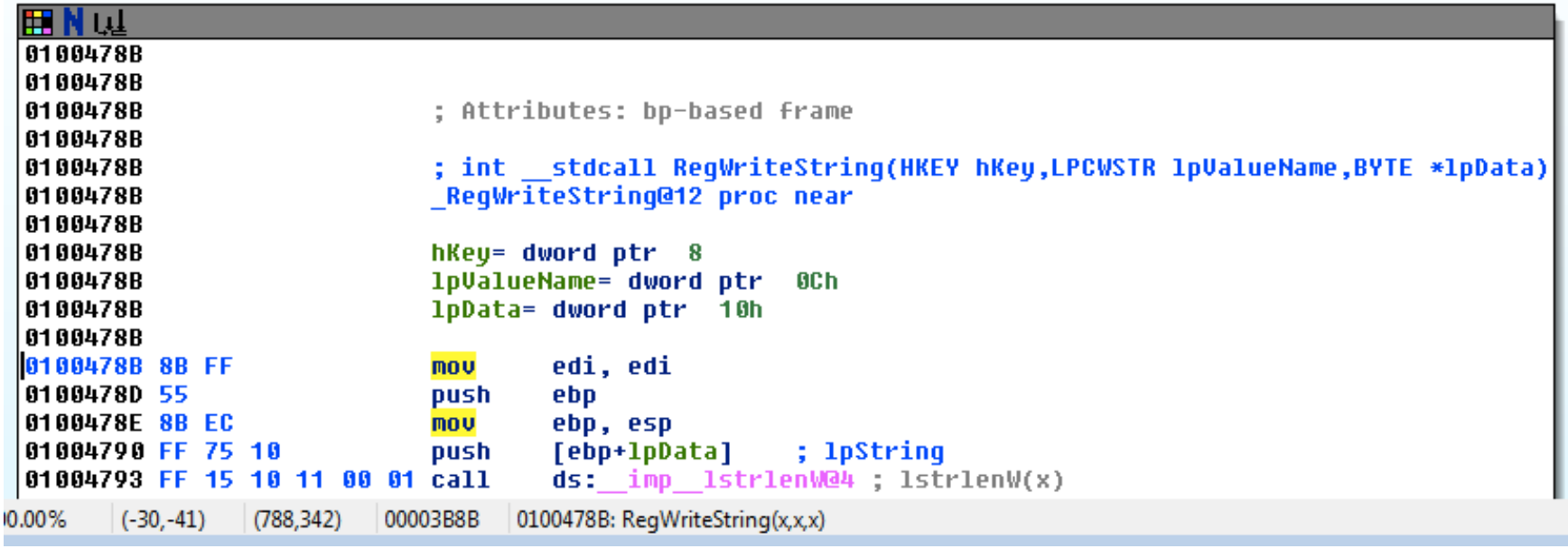
- Double-click function
- Jump to code in other views





Function Call

- Parameters pushed onto stack
- CALL to start function



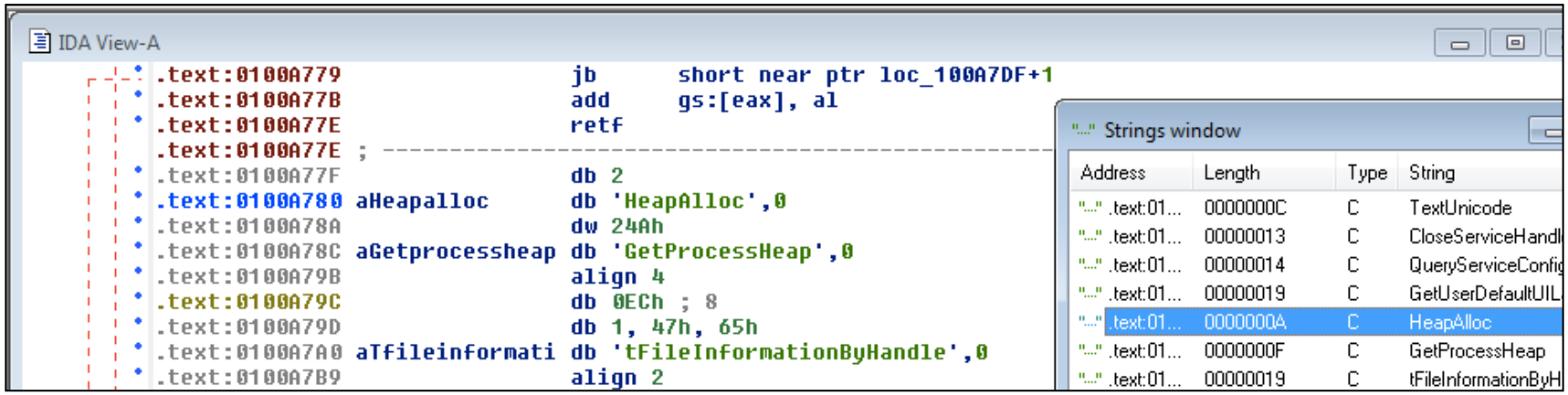
Returning to the Default View

- Windows, Reset Desktop
- Windows, Save Desktop
  - To save a new view

Navigating IDA Pro

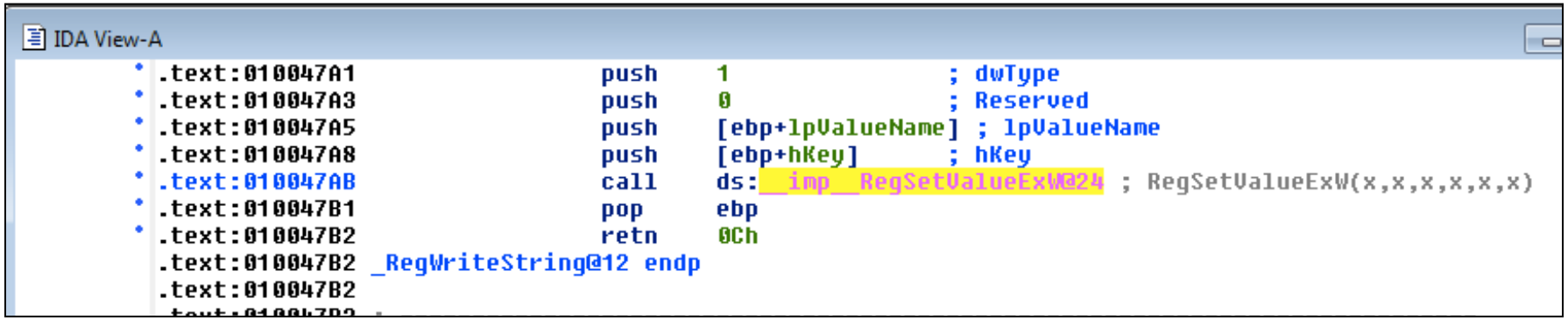
Imports or Strings

- Double-click any entry to display it in the disassembly window



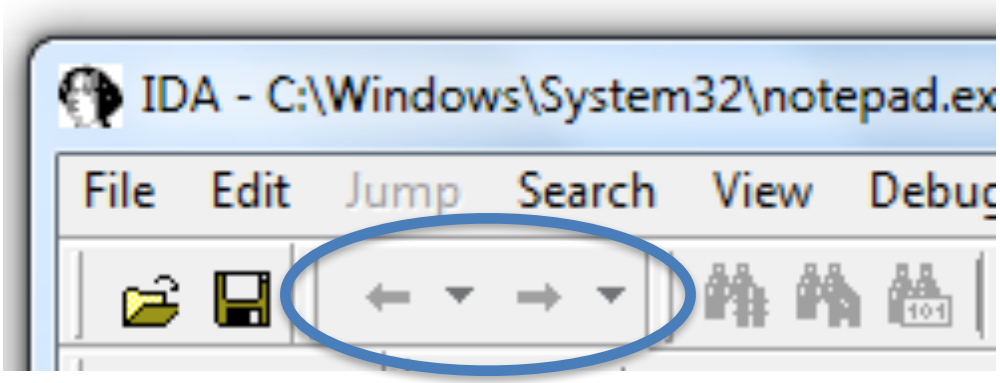
Using Links

- Double-click any address in the disassembly window to display that location

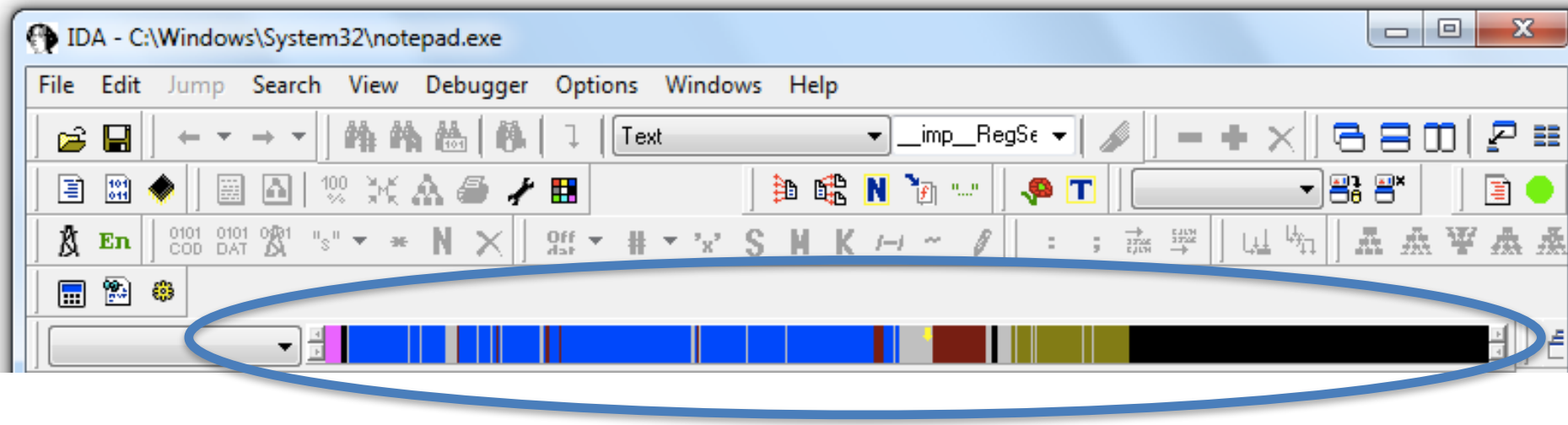


History

- Forward and Back buttons work like a Web browser



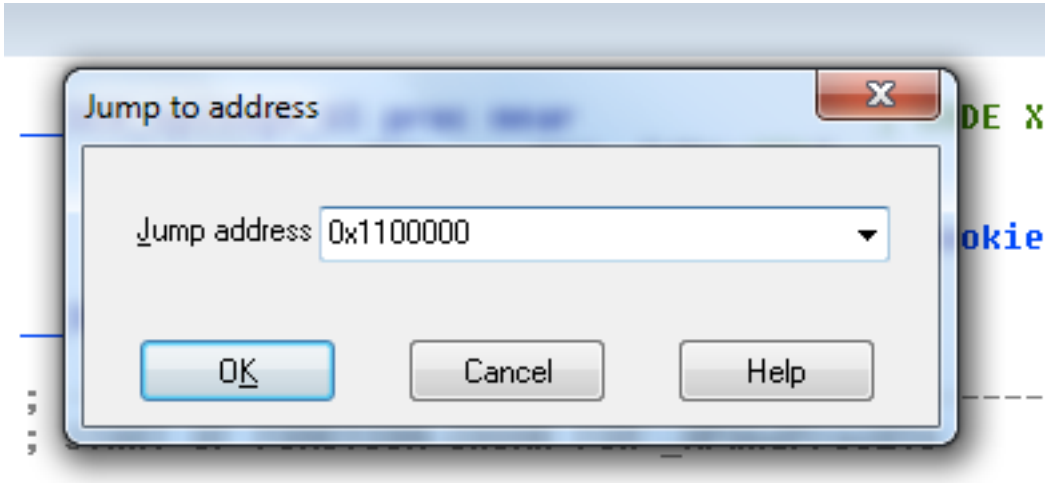
Navigation Band



- **Light blue**: Library code
- **Red**: Compiler-generated code
- **Dark blue**: User-written code – **Analyze this**

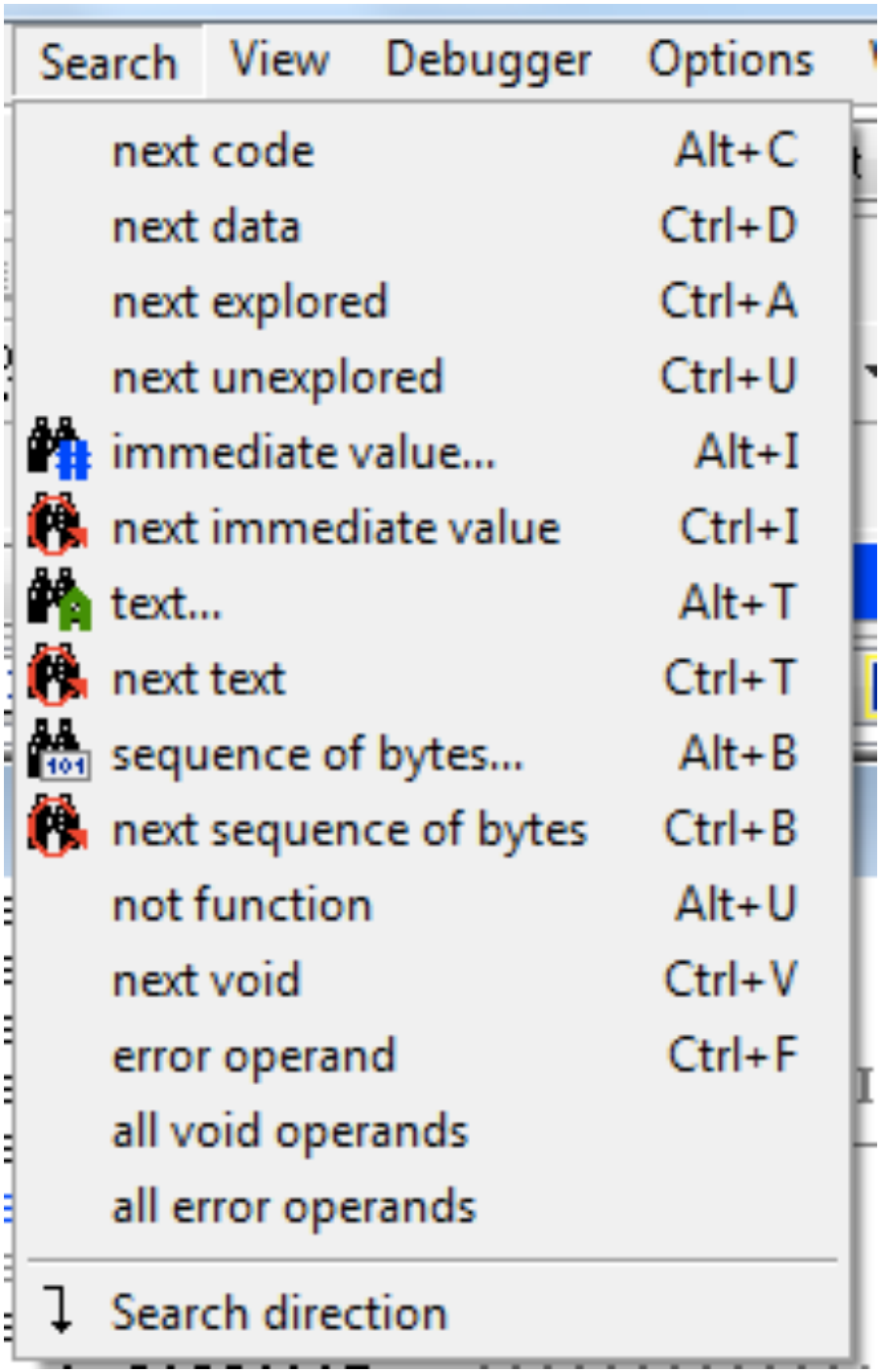
Jump to Location

- Press **G**
- Can jump to address or named location



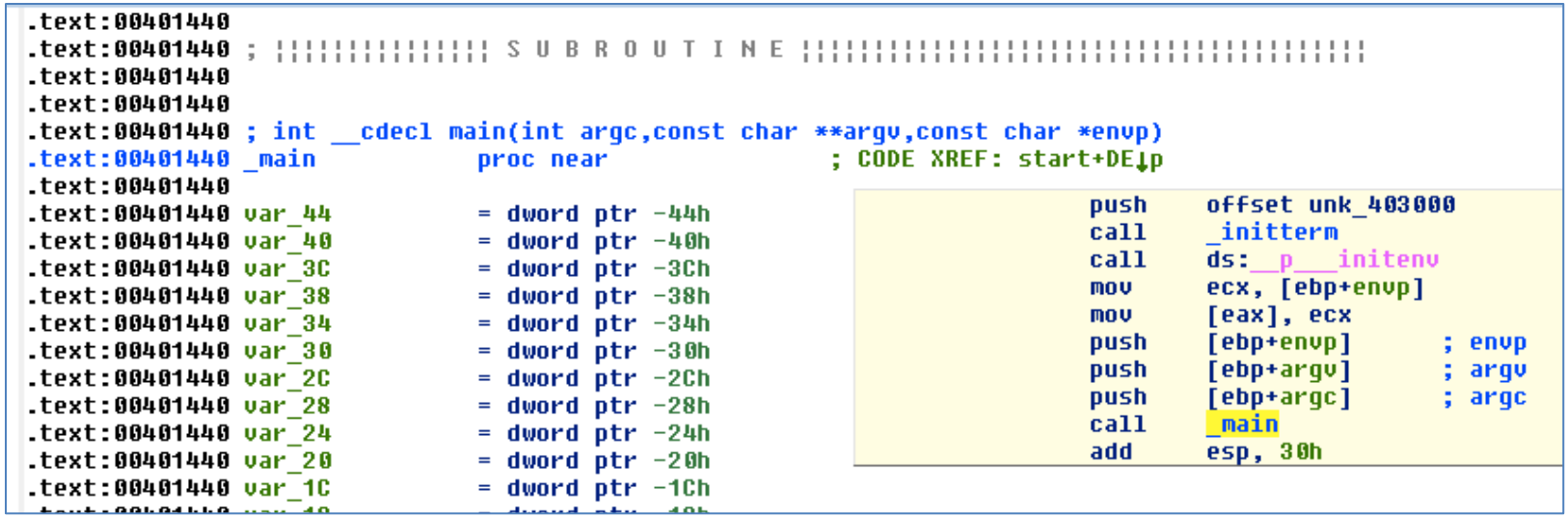
Searching

- Many options
- Search, Text is handy



Using Cross-References

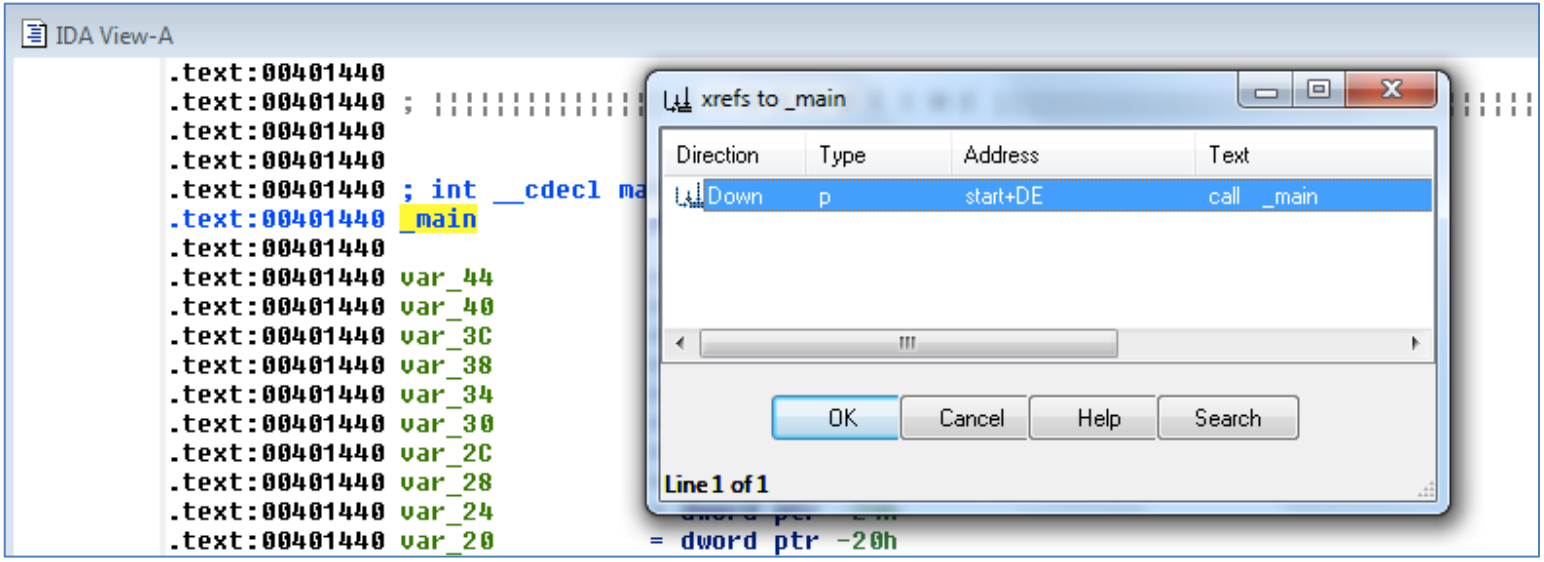
Code Cross-References



- XREF comment shows where this function is called
- But it only shows a couple of cross-references by default

To See All Cross-References

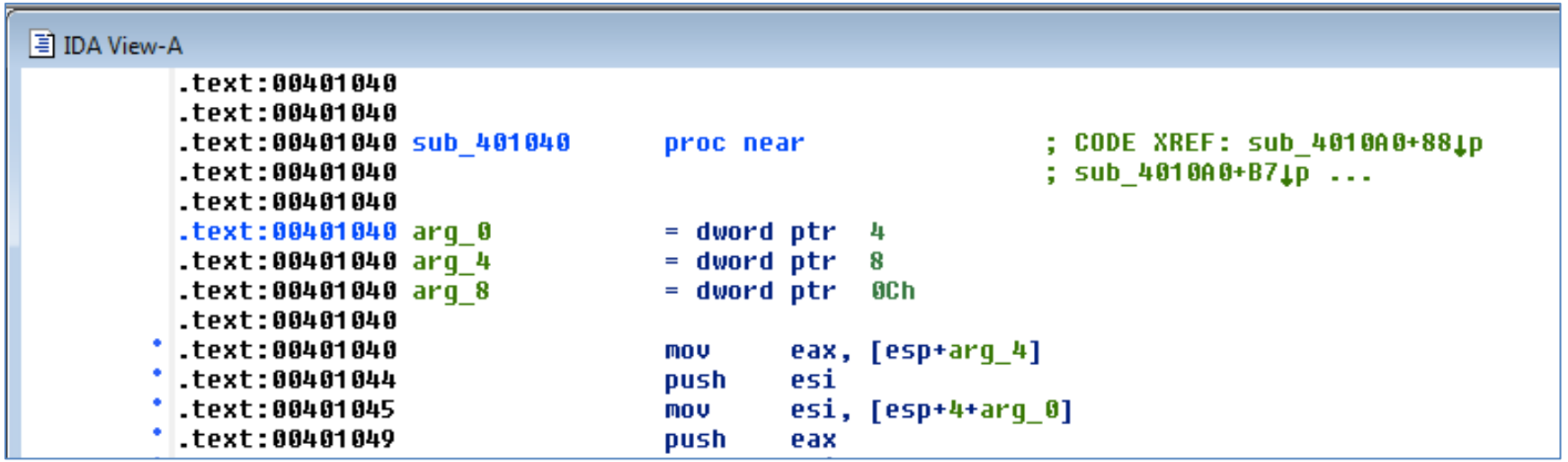
- Click function name and press **X**



Analyzing Functions

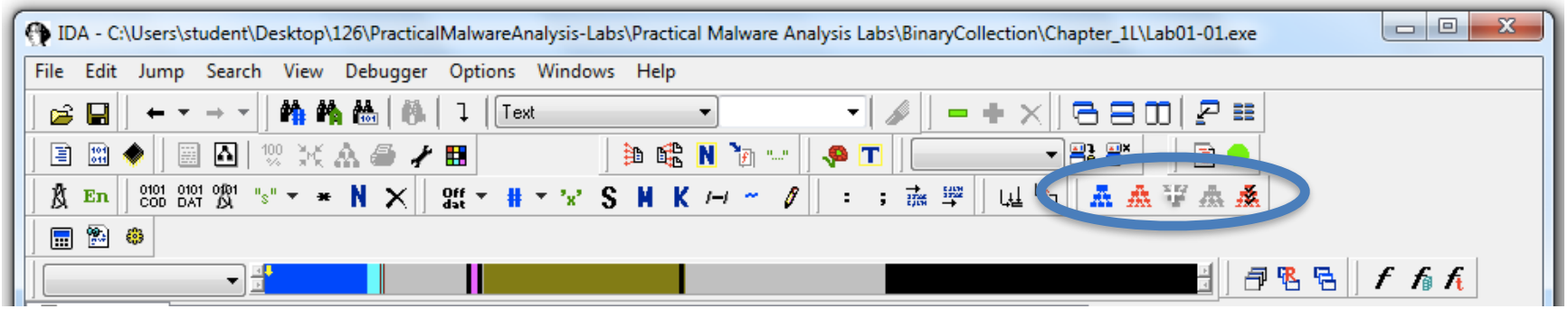
Function and Argument Recognition

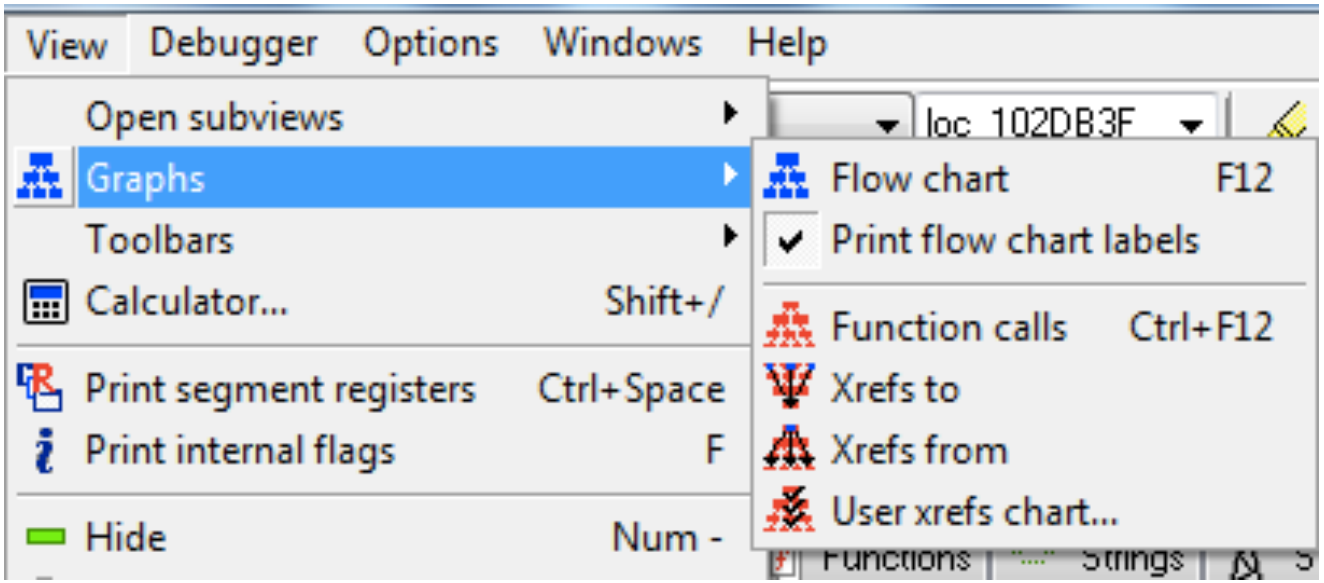
- IDA Pro identifies a function, names it, and also names the local variables
- It's not always correct



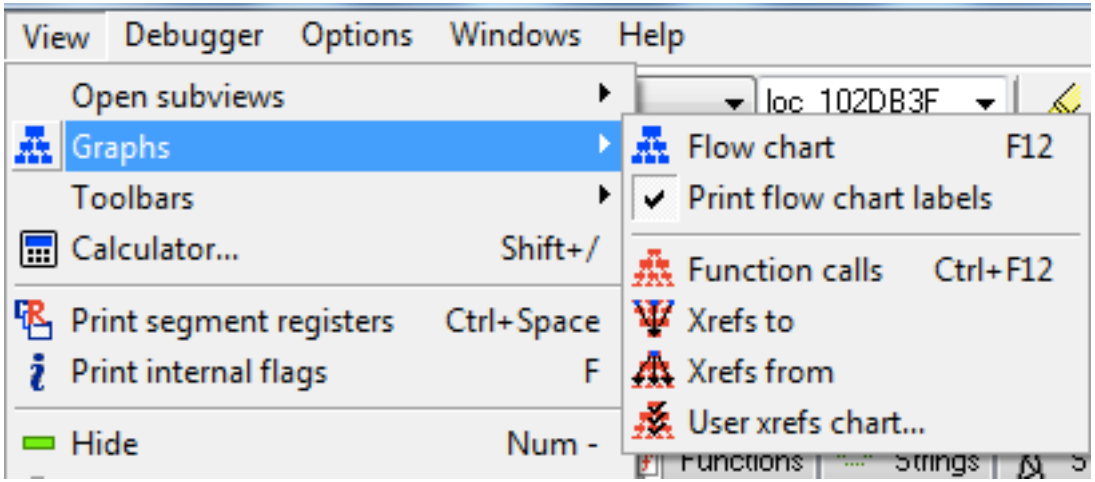
Using Graphing Options

Graphing Options



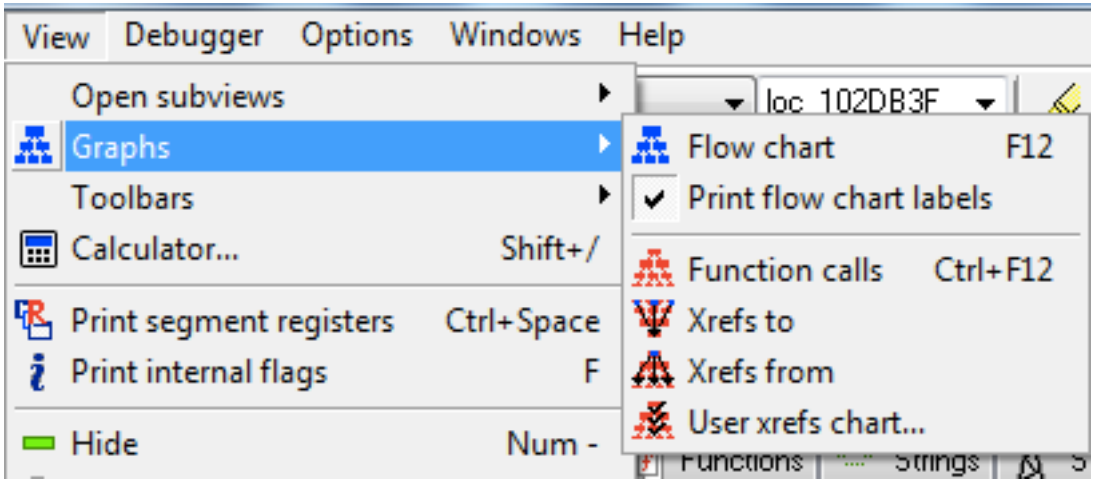


Graphing Options



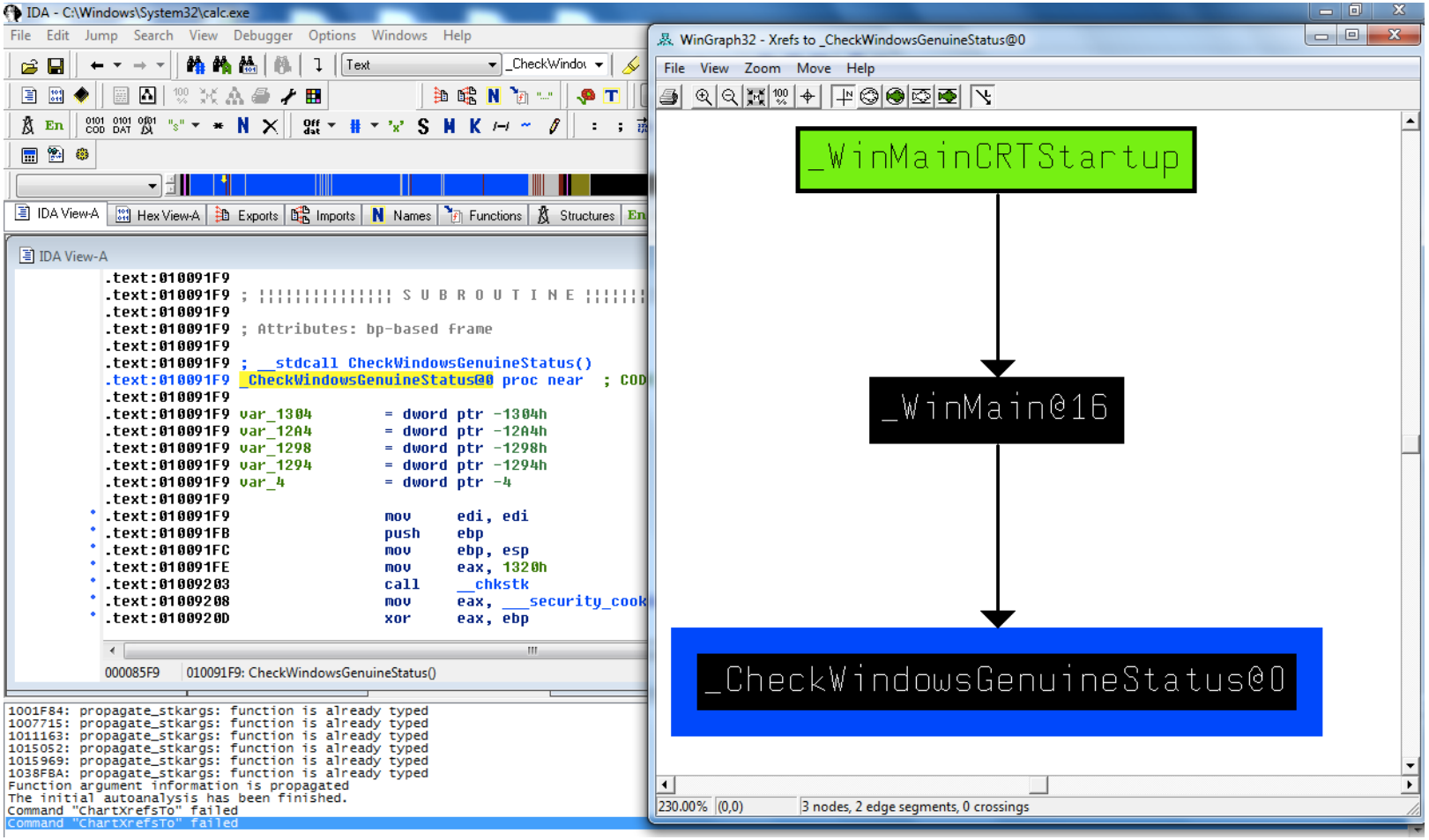
- These are "Legacy Graphs" and cannot be manipulated with IDA
- The first two seem obsolete
  - **Flow chart**
    - Create flow chart of current function
  - **Function calls**
    - Graph function calls for entire program

Graphing Options

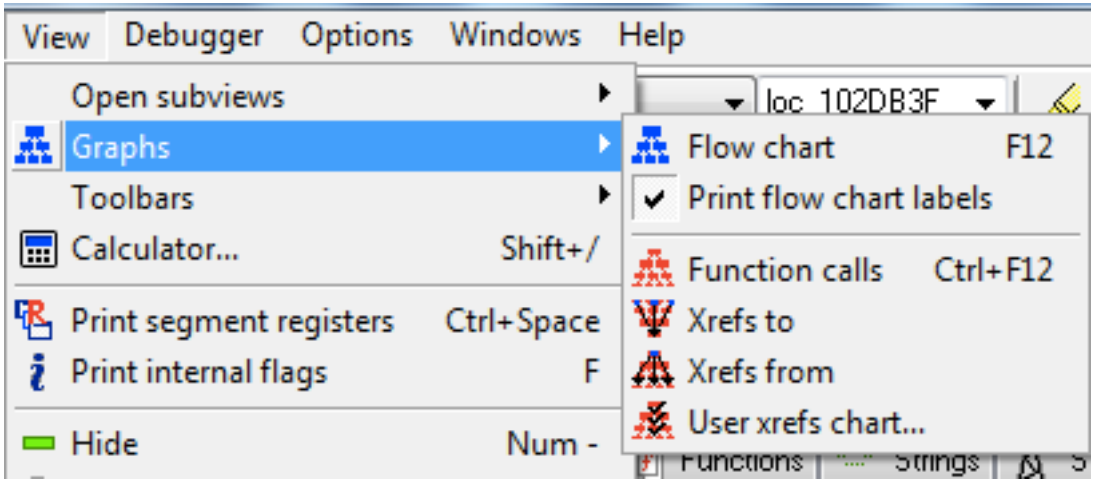


- **Xrefs to**
  - Graphs XREFs to get to selected XREF
  - Can show all the paths that get to a function

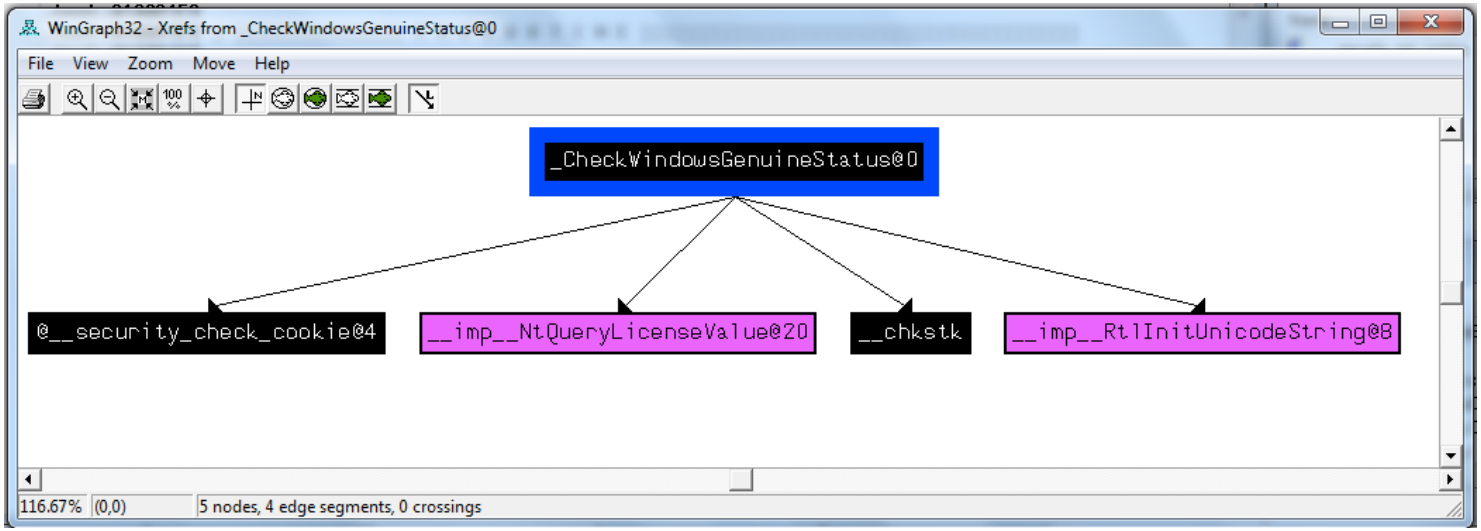
Windows Genuine Status in Calc.exe



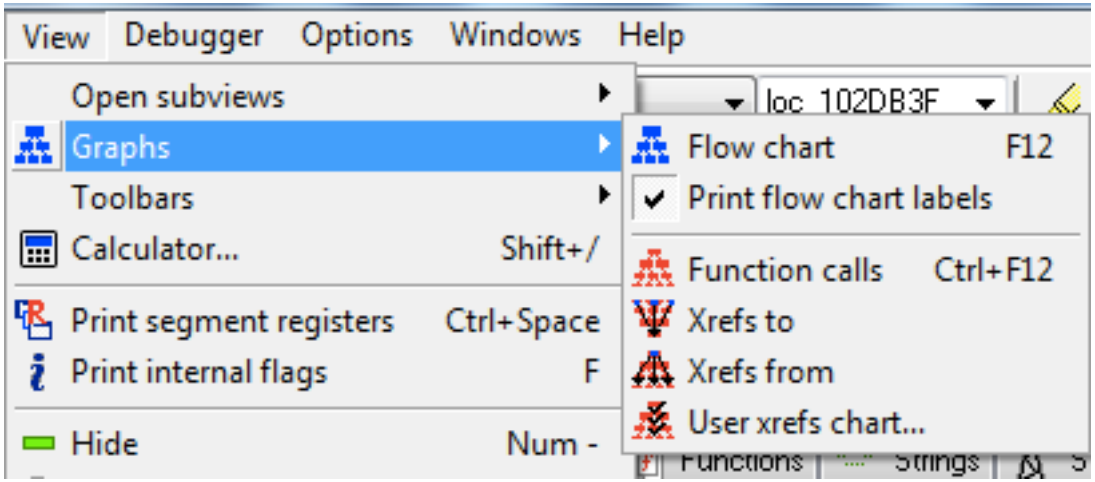
Graphing Options



- **Xrefs from**
  - Graphs XREFs from selected XREF
  - Can show all the paths that exit from a function



Graphing Options



- **User xrefs chart...**
  - Customize graph's recursive depth, symbols used, to or from symbol, etc.
  - The only way to modify legacy graphs

Enhancing Disassembly

Warning

- There's no Undo, so if you make changes and mess them up, you may be sorry

Renaming Locations

- You can change a name like **sub_401000** to **ReverseBackdoorThread**
- Change it in one place, IDA will change it everywhere else

Table 6-2. Function Operand Manipulation

Without renamed arguments

```
004013C8  mov   eax, [ebp+arg_4]
004013CB  push  eax
004013CC  call  _atoi
004013D1  add   esp, 4
004013D4  mov [ebp+var_598], ax
004013DB  movzx ecx, [ebp+var_598]
004013E2  test  ecx, ecx
004013E4  jnz   short loc_4013F8
004013E6  push  offset aError
004013EB  call  printf
004013F0  add   esp, 4
004013F3  jmp   loc_4016FB
004013F8 ; ----------------------
004013F8
004013F8 loc_4013F8:
004013F8  movzx edx, [ebp+var_598]
004013FF  push  edx
00401400  call  ds:htons
```

With renamed arguments

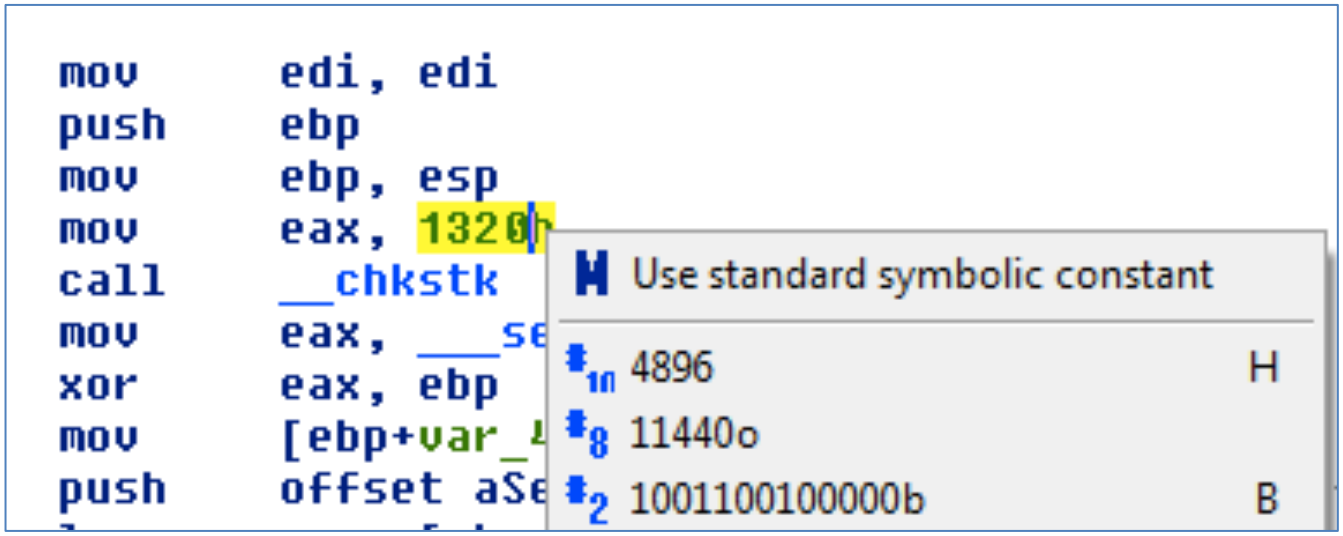
```
004013C8  mov   eax, [ebp+port_str]
004013CB  push  eax
004013CC  call  _atoi
004013D1  add   esp, 4
004013D4  mov   [ebp+port], ax
004013DB  movzx ecx, [ebp+port]
004013E2  test  ecx, ecx
004013E4  jnz   short loc_4013F8
004013E6  push  offset aError
004013EB  call  printf
004013F0  add   esp, 4
004013F3  jmp   loc_4016FB
004013F8 ; ----------------------
004013F8
004013F8 loc_4013F8:
004013F8  movzx edx, [ebp+port]
004013FF  push  edx
00401400  call  ds:htons
```

Comments

- Press colon (:) to add a single comment
- Press semicolon (;) to echo this comment to all Xrefs

Formatting Operands

- Hexadecimal by default
- Right-click to use other formats



Using Named Constants

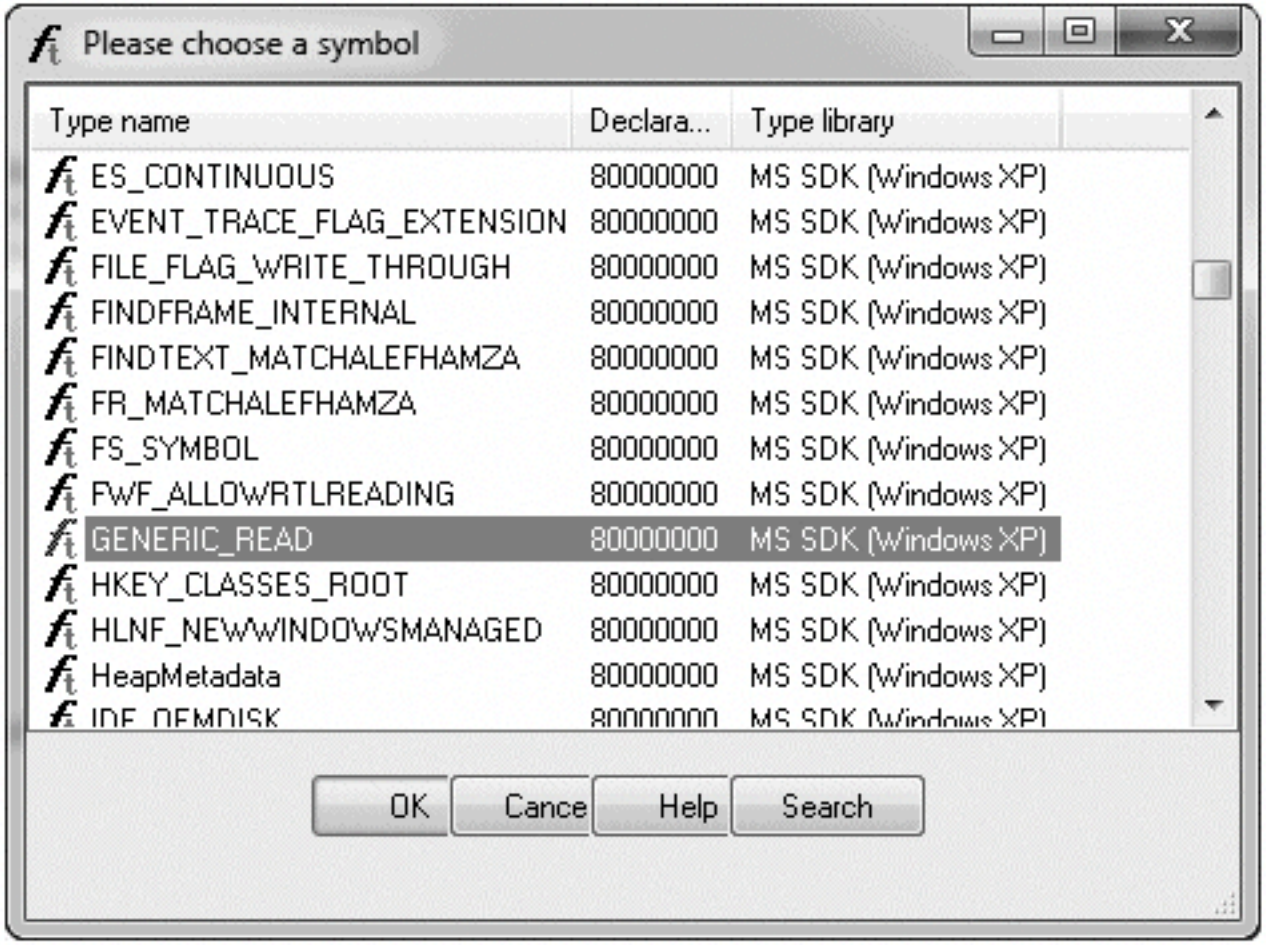


Figure 6-11. Standard symbolic constant window

Using Named Constants

- Makes Windows API arguments clearer

Before symbolic constants

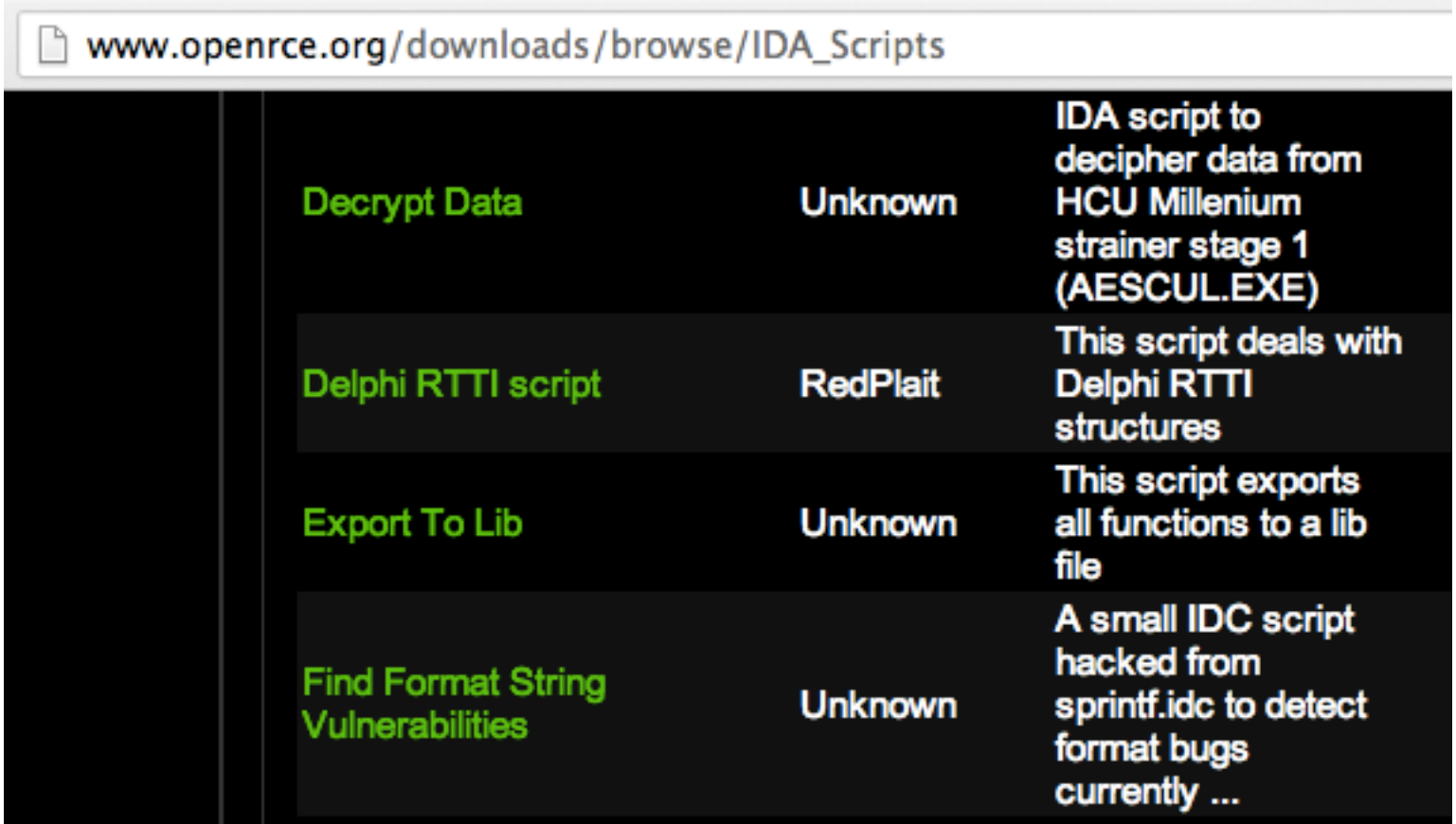
```
mov     esi, [esp+1Ch+argv]
mov     edx, [esi+4]
mov     edi, ds:CreateFileA
push    0    ; hTemplateFile
push    80h  ;
dwFlagsAndAttributes
push    3    ;
dwCreationDisposition
push    0    ;
lpSecurityAttributes
push    1    ; dwShareMode
```

After symbolic constants

```
mov     esi, [esp+1Ch+argv]
mov     edx, [esi+4]
mov     edi, ds:CreateFileA
push    NULL  ; hTemplateFile
push    FILE_ATTRIBUTE_NORMAL ;
dwFlagsAndAttributes
push    OPEN_EXISTING         ;
dwCreationDisposition
push    NULL                  ;
lpSecurityAttributes
push    FILE_SHARE_READ       ; dwShareMode
```

Extending IDA with Plug-ins

- IDC (IDA's scripting language) and Python scripts available (link Ch 6a)




Kahoot!