



Hex-Rays

<http://www.hex-rays.com>

Email: info@hex-rays.com

Tel: +32-4-222-4300

Fax: +32-4-223-5600

Hex-Rays decompiler v1.0



Are you a software security provider, anti-virus company, vulnerability research company, military or government organization working on software validation and verification? Our binary analysis tool Hex-Rays can cut your analysis times by an order of magnitude or more!

Hex-Rays is a decompiler that transforms binary applications into a high level readable text. Unlike disassemblers, which perform the same task at a lower level, the decompiler output is concise, closer to the standard way programmers use to write applications. This alone can save hours of work because analysts mentally map the disassembly output to high-level concepts. Decompiler frees them of this routine and boring task. Since the decompiler output is similar to high level languages, any regular C/C++ programmer can understand it.

Hex-Rays is the first decompiler that can handle real world applications. It embodies more than ten years of proprietary research and implements unpublished algorithms and innovative ideas. Its output is clean, well structured, and easily modifiable.

Hex-Rays comes with one year of technical support by email, free bugfixes and updates. At the end of the support period you will be entitled to a substantial discount for the next support period.

Facts about Hex-Rays:

- The decompiler supports 32-bit compiler generated x86 code
- It can handle code generated by any mainstream C/C++ compiler
- It is very fast. Most functions are analyzed instantaneously
- It has interactive and batch modes
- It is shipped as an **IDA Pro** plugin. IDA 5.1 or higher is required to run it
- Floating point instructions (as well as XMM/MMX/SSE* instructions) are not supported in the current version
- Exception handling is not supported in the current version
- Since decompilation in general is an unsolvable problem, the output is not 100% reliable (which is the case with disassemblers as well)

1 Hexadecimal code: incomprehensible for humans

```
00 00 C6 05 85 14 00 00 02 C6 05 89 14 00 00 00
74 28 A1 78 14 00 00 6A 08 50 E8 F5 10 00 00 83
C4 08 83 F8 FF 75 02 33 C0 83 E0 0F 0F B6 CB C1
E0 08 0B C1 A3 90 14 00 00 C3 0F B6 C3 3D 80 00
00 00 A3 90 14 00 00 72 0A 05 00 0F 00 00 A3 90
14 00 00 C3 53 8B D8 C1 F8 0A 83 F8 03 0F 87 C8
01 00 00 FF 24 85 90 05 00 00 83 FB 40 75 0B 66
C7 05 7C 14 00 00 04 00 5B C3 85 DB 75 0B 66 C7
05 7C 14 00 00 0D 00 5B C3 83 FB 04 75 0B 66 C7
05 7C 14 00 00 1A 00 5B C3 83 FB 02 75 0B 66 C7
05 7C 14 00 00 24 00 5B C3 83 FB 03 75 0B 66 C7
05 7C 14 00 00 21 00 5B C3 F7 C3 F8 0F 00 00 75
11 8B C3 66 C7 05 7C 14 00 00 25 00 5B E9 B2 FE
FF FF F7 C3 80 0F 00 00 75 1F 8B C3 C1 F8 05 83
E0 03 66 8B 0C 45 D0 05 00 00 8B C3 66 89 0D 7C
14 00 00 5B E9 8B FE FF FF 8B D3 C1 FA 06 66 8B
04 55 B0 05 00 00 66 A3 7C 14 00 00 8B C3 E8 71
FE FF FF C1 EB 05 83 E3 01 66 89 1D A2 14 00 00
C6 05 9D 14 00 00 01 C6 05 A1 14 00 00 00 5B C3
8B CB C1 F9 08 83 E1 03 66 8B 14 4D A8 05 00 00
8B C3 66 89 15 7C 14 00 00 E8 36 FE FF FF C1 FB
05 83 E3 07 89 1D A4 14 00 00 C6 05 9D 14 00 00
05 C6 05 A1 14 00 00 00 5B C3 8B C3 C1 F8 08 83
E0 03 83 E8 00 74 79 83 E8 01 6A 0B 74 3A A1 78
14 00 00 50 E8 8B 0F 00 00 83 E0 60 81 E3 FF 01
00 00 83 C4 08 C1 E0 04 0B C3 66 C7 05 7C 14 00
00 1B 00 C6 05 85 14 00 00 07 A3 90 14 00 00 C6
05 89 14 00 00 09 5B C3 8B 0D 78 14 00 00 51 E8
50 0F 00 00 83 E0 60 0F B6 D3 83 C4 08 C1 E0 04
0B C2 66 C7 05 7C 14 00 00 19 00 C6 05 85 14 00
```

2 Disassembler output: makes sense but lengthy

```
mov     eax, dword ptr ds:?cmd@@@3Vi
add     eax, 1
push   1                ; int
push   eax              ; unsigned
call   ?get_flags_ex@@YAKKH@Z ; ge
mov     ecx, eax
add     esp, 8
and     ecx, 600h
jz     short loc_2D1
cmp     ecx, 200h
jnz    short loc_2DE

; CODE XREF
shr     eax, 0Ch
test   al, 1
jnz    short loc_2DE
mov     eax, 1
retn

; CODE XREF
; may_grow(
xor     eax, eax
retn
```

3 Decompiler output: concise and familiar for programmers

```
if ( update )
{
    result = "+-";
}
else
{
    if ( add )
    {
        result = "+";
    }
    else
    {
        if ( dword_41F
```

For more information about Hex-Rays please visit our web site: <http://www.hex-rays.com>