

Forum:[Hack Forums](#)

Thread: [Converting Hex String to Hex DWORD](#)

unknown2me - January 18, 2015, 5:55 am

I've tried possibly everything.

e.g. string = "0x1337";

I'll use strtol or strtoul to convert the string to DWORD but I get the decimal equivalent. I've also tried using std::hex but DWORD is assigned the decimal value.

String = "0x1337", DWORD = 4919

I want DWORD to = 1337

(Needed for RPM)

Mr. Dr. Prof. Patrick - January 18, 2015, 6:10 am

just create your own atoi function... this thread and the one below it.. they worry me...cmon these things are super simple

Nichole - January 18, 2015, 7:03 am

It's all binary data. How you display that data is up to you.

```
Code int x = 4919;
printf("0x%X\n",x);//Prints 0x1337
printf("%i\n",x);//Prints 4919
```

unknown2me - January 18, 2015, 7:12 am

(Jan 18, 2015 08:03 AM)Nichole Wrote: [\[CITE\]](#) It's all binary data. How you display that data is up to you.

```
Code int x = 4919;
printf("0x%X\n",x);//Prints 0x1337
printf("%i\n",x);//Prints 4919
Need to store it in DWORD variable to be used further via ReadProcessMemory
```

Nichole - January 18, 2015, 7:14 am

(Jan 18, 2015 08:12 AM)unknown2me Wrote: [\[CITE\]](#) Need to store it in DWORD variable to be used further via ReadProcessMemory

A DWORD is just a typedef for unsigned long..

```
Code DWORD dwVar = 0x1337;
```

unknown2me - January 18, 2015, 8:02 am

(Jan 18, 2015 08:14 AM)Nichole Wrote: [\[CITE\]](#) A DWORD is just a typedef for unsigned long..

```
Code DWORD dwVar = 0x1337;
```

I'm aware... But i'm trying to make my DwORD = "0x1337" via the string

not assigning the value itself.

Kenshin13 - January 18, 2015, 8:16 am

There's an old deprecated function called "itoa" you could use.

<http://www.cplusplus.com/reference/cstdlib/itoa/>

Nichole - January 18, 2015, 8:25 am

(Jan 18, 2015 09:02 AM)unknown2me Wrote: [\[CITE\]](#) I'm aware... But i'm trying to make my DWORD = "0x1337" via the string

not assigning the value itself.

0x1337 is just a binary number in base 16.

4919 is just a binary number in base 10.

0001001100110111 is just a binary number in base 2.

```
Code int main()
{

    char* szVal = "0x1337";
    DWORD dwVal = 0;

    printf("%s\n",szVal);

    dwVal = strtol(szVal,0,0);

    printf("%i\t0x%X\n",dwVal,dwVal);

    return 0;
}
```

Edit: Minor change on the base 2.

g4143 - January 18, 2015, 9:18 am

Did you try something like this?

```
Code #include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {

    char * str = "0x56f67";

    long int ans = strtol(str, NULL, 16);

    fprintf(stdout, "ans->%d\n", ans);

    return (EXIT_SUCCESS);
}
```

rooot - January 18, 2015, 9:32 am

Considering it's a string. 0x1337 is ASCII. Convert the ASCII representation to Binary then Binary into Hex.

Which isn't hard at all.

Considering Binary is base 2, 11011 in binary would be 27 (Base 10)
Considering 4 bits are a representation of a 'char' let's split this up. 0001|1011 (leading zeros)
1011 = 11
1 = 1 so 0x1B

unknown2me - January 18, 2015, 9:45 am

Think I'm being misunderstood. I need to take the value FROM the string and then ASSIGN the value to DWORD.

```
String value = "0x1337"; //Value wont be assigned like this
DWORD importValue;
```

I don't want to print the value; I want to assign it to my DWORD variable.

0xDEAD10CC - January 18, 2015, 10:31 am

Is this C or C++? Everyone above seemed to assume that it was C yet none of that was even specified. Here's how you get the value of 1337 from the hex string "0x1337" into a DWORD variable in C++:

```
Code #include <iostream>
#include <windows.h>
#include <sstream>

int main()
{
    DWORD val;
    char str[] = "0x1337"; // DWORD = 4919
    std::istringstream isstrm(str+2);
    isstrm >> val; // provide check for success/failure here
    std::cout << val << '\n'; // value = 1337
}
```

If you want the hexadecimal value itself to be stored in the dword (4919 (which is the same as 0x1337) and not 1337 (in decimal)) then:

```
Code #include <iostream>
#include <windows.h>
#include <sstream>

int main()
{
    DWORD val;
    char str[] = "0x1337"; // DWORD = 4919
    std::istringstream isstrm(str);
    isstrm >> std::hex >> val;
    std::cout << val << '\n'; // value = 4919
}
```

rooot - January 18, 2015, 10:58 am

^0xDEAD is it just me or are you guys purposely fucking around with those code tags?
It seems to me they have weird characters. I can view the code when i'm quoting.

0xDEAD10CC - January 18, 2015, 1:02 pm

(Jan 18, 2015 11:58 AM)rooot Wrote: [\[CITE\]](#) ^0xDEAD is it just me or are you guys purposely fucking around with those code tags?
It seems to me they have weird characters. I can view the code when i'm quoting.

It's just you unless you don't understand C++ syntax.. Reinstall or try another browser.

rooot - January 18, 2015, 1:09 pm

(Jan 18, 2015 02:02 PM)0xDEAD10CC Wrote: [\[CITE\]](#) It's just you unless you don't understand C++ syntax.. Reinstall or try another browser.

It's weird, it was doing it in incognito mode. I guess it was the browser. It was completely as if you guys were trolling OP. Seemed quite funny until i hit 'quote' and realized it was the browser

0xDEAD10CC - January 18, 2015, 1:20 pm

@OP - as a function like you wanted via PM:

```
Code DWORD convert(const std::string &str)
{
    DWORD val;
    std::istream isstrm(str.substr(2));
    isstrm >> val; // provide check for success/failure here
    return val;
}
```

Use:

```
Code DWORD val = convert("0x1337");
```

Next time don't PM me if it's something related to a thread. I wrote this comment for a reason too:

```
Code // provide check for success/failure here
```

This as a whole:

```
Code (isstrm >> val)
```

returns something useful in itself.