

Forum:[Hack Forums](#)

Thread: [Hardware ID Generator \[HWID\] \[Revamped & Faster\]](#)

Malum™ - December 30, 2016, 7:40 am

Was disassembling someone's program and noticed there HWID method and was severely disappointed. Lol. So I Google'd around and found [this](#). Which works just fine but Jesus fucking Christ was it slow. So I re-coded it and organized it just a tad bit more. I haven't tested it on multiple computers, just my own.

Notes: Make sure to [Add Reference](#) System.Management

Because I used a translator from C# to VB.NET. They output different HWID. EX: C# 1234-4321 / VB 4321-1234

Benchmarked at 100ms for VB.NET and 0ms for C#.

Added [Using Statements](#) to properly Dispose of ManagementObjects.

Increased speed using [ManagementObjectSearcher](#)

Properly disposed of objects thanks to [Ch@rLi3 \(3D?tD3v\)](#)

Fixed [ManagementObjectSearcher](#) error. Stupid error. My bad.

Think I optimized the WMI usage a little more. 1/9/2017

Added using statement to SHA1 hash and changed to static class thanks to [Ninja](#). 1/9/2017

[aneeshkollam5](#) found a bug with External Hard Drives and possibly USB devices. [This](#) looks promising. 1/11/2017

Code Console.WriteLine(HardwareID.Generate())

[C# PasteBin Link](#)

[VB.NET PasteBin Link](#)

Benchmark (C#)

Spoiler [\(Click to View\)](#)

Please give me suggestions on how to make it faster or better! Politely please. :)

passwords - December 30, 2016, 7:42 am

Nice & simple.. Just how I like it!

Ninja - December 30, 2016, 9:08 am

You could make a static class that returns a struct with properties (like each segment) and different ToString functions similar to the GUID class.

Malum™ - December 30, 2016, 9:23 am

(Dec 30, 2016 10:08 AM)Ninja Wrote: [\[CITE\]](#) You could make a static class that returns a struct with properties (like each segment) and different ToString functions similar to the GUID class.

That's a decent idea. I don't know about the ToString functions but returning a struct with each bit of info would be nice. If that is what you mean.

Ninja - December 30, 2016, 9:33 am

(Dec 30, 2016 10:23 AM)Malum™ Wrote: [\[CITE\]](#) That's a decent idea. I don't know about the ToString functions but returning a struct with each bit of info would be nice. If that is what you mean.

ToString can return different formats such as hyphens, no hyphens, encapsulated by curly brackets, etc. Look into the GUID ToString functions.

calvv - December 30, 2016, 9:36 am

Good job! Adding this to my bookmarks ?

Malum™ - December 30, 2016, 9:38 am

(Dec 30, 2016 10:33 AM)Ninja Wrote: [\[CITE\]](#) ToString can return different formats such as hyphens, no hyphens, encapsulated by curly brackets, etc. Look into the GUID ToString functions.

Oh, gotcha, I'll play with it later and eventually update the thread by tonight with those kind of features.

+Haze - December 30, 2016, 11:34 am

You should really work on the meaningfulness of your code. What I mean is that some of your methods and variables names are not so descriptive in the context they're used in.

Other than that good job on the release

Zilog80 - December 30, 2016, 11:50 am

Pro-tip:

Not in your code, but try to avoid using With ... End With

Why?

Because this:

When debugging, VS intellisense can not show you the value inside With block, but the intellisense works fine outside With block.

Seems a nonsense, but is really helpful when you are debugging and don't need switching to watch-list window.

The code provided is an example, not a real world imba algorithm, but hope you get the idea

Malum™ - December 30, 2016, 11:55 am

(Dec 30, 2016 12:34 PM)+Haze Wrote: [\[CITE\]](#) You should really work on the meaningfulness of your code. What I mean is that some of your methods and variables names are not so descriptive in the context they're used in.

Other than that good job on the release
As in commenting the code or what? Give me some examples if you would.

(Dec 30, 2016 12:50 PM)Zilog80 Wrote: [\[CITE\]](#) Pro-tip:

Not in your code, but try to avoid using With ... End With
I already knew that. I just took advantage of the feature when I used VB.NET otherwise never touch it. Thank you for the warning though!

sh@rp - December 30, 2016, 2:38 pm

Ohhh Damn..... don't forget to dispose objects.....

```
Code using (ManagementObjectCollection moc = new ManagementClass(wclass).GetInstances()) {  
    //.....  
    //.....  
}  
but ... nice try. ;)
```

Malum™ - December 30, 2016, 7:52 pm

(Dec 30, 2016 03:38 PM)Ch@rLi3 (3D?tD3v) Wrote: [\[CITE\]](#) Ohhh Damn..... don't forget to dispose objects.....

```
Code using (ManagementObjectCollection moc = new ManagementClass(wclass).GetInstances()) {  
    //.....  
    //.....  
}  
but ... nice try. ;)
```

"The using statement ensures that Dispose is called even if an exception occurs while you are calling methods on the object." - [MSDN](#)

Unless you have other reasoning?

EDIT: Oh! I see what I missed. Hopefully GC will have my back. Haha. I'll fix it when I get home.

ubbelol - December 31, 2016, 3:08 am

(Dec 30, 2016 08:40 AM)Malum™ Wrote: [\[CITE\]](#) Please give me suggestions on how to make it faster or better! Politely please. :)

If you want to speed it up then don't use WMI.

sh@rp - December 31, 2016, 8:48 am

(Dec 30, 2016 08:52 PM)Malum™ Wrote: [\[CITE\]](#) Unless you have other reasoning?

EDIT: Oh! I see what I missed. Hopefully GC will have my back. Haha. I'll fix it when I get home.
If you want to give advices and lessons please reads properly MSDN :

Quote:As a rule, when you use an IDisposable object, you should declare and instantiate it in a using statement. The using statement calls the Dispose method on the object in the correct way, and (when you use it as shown earlier) it also causes the object itself to go out of scope as soon as Dispose is called. Within the using block, the object is read-only and cannot be modified or reassigned.

Does your "moc" object isn't disposable and also does your instance is used outside of your

GetHardwareProperty method

ok ... nstr...

.....

Malum™ - December 31, 2016, 6:10 pm

(Dec 31, 2016 09:48 AM)Ch@rLi3 (3D?tD3v) Wrote: [\[CITE\]](#) If you want to give advices and lessons please reads properly MSDN :

Does your "moc" object isn't disposable and also does your instance is used outside of your GetHardwareProperty method

No need to be sassy. Lol. I misunderstood and did not read further into it. It was a misunderstanding. Thank you for the information!

(Dec 31, 2016 04:08 AM)ubbelol Wrote: [\[CITE\]](#) If you want to speed it up then don't use WMI. Do you know a different alternative? I tried looking into EventLogs but didn't quite understand it.

sh@rp - January 1, 2017, 3:39 am

(Dec 31, 2016 07:10 PM)Malum™ Wrote: [\[CITE\]](#) Do you know a different alternative? I tried looking into EventLogs but didn't quite understand it.

plInvoke functions (<http://www.pinvoke.net/>) or registry directly :

CPU :

getnativesysteminfo

....

.....

DEVICES :

SetupDiEnumDeviceInterfaces

SetupDiGetDeviceInterfaceDetail

DeviceIoControl

CreateDevice

CM_Get_Device_ID

SetupDiOpenDeviceInfo

SetupDiGetDeviceRegistryProperty

...

.....

MONITORS :

EnumDisplayMonitors

GetMonitorInfo

..

.....

.....asO....

Malum™ - January 1, 2017, 3:43 am

(Jan 1, 2017 04:39 AM)Ch@rLi3 (3D?tD3v) Wrote: [\[CITE\]](#) plInvoke functions (<http://www.pinvoke.net/>) or registry directly :

I figured out just by using a query instead of through every property I cut the time down to basically 0ms except for VB.NET for some reason.. So this method will do just fine for now :)

Xogg - January 7, 2017, 10:13 pm

My nigga always releasing HQ shit.

Good work man!

Sn0wX - January 8, 2017, 6:53 pm

i wont comment on your shower of shit wmi abuse. As for string builder class. You ever read its documentation? No you have not....

Although the StringBuilder class generally offers better performance than the String class, you should not automatically replace String with StringBuilder whenever you want to manipulate strings. Performance depends on the size of the string, the amount of memory to be allocated for the new string, the system on which your app is executing, and the type of operation. You should be prepared to test your app to determine whether StringBuilder actually offers a significant performance improvement.

this site is a bloody joke.

D9ni3I - January 9, 2017, 3:00 pm

simple and very useful, thanks you :)

Malum™ - January 9, 2017, 5:59 pm

(Jan 8, 2017 07:53 PM)Sn0wX Wrote: [CITE](#) i wont comment on your shower of shit wmi abuse. As for string builder class. You ever read its documentation? No you have not....

Although the StringBuilder class generally offers better performance than the String class, you should not automatically replace String with StringBuilder whenever you want to manipulate strings. Performance depends on the size of the string, the amount of memory to be allocated for the new string, the system on which your app is executing, and the type of operation. You should be prepared to test your app to determine whether StringBuilder actually offers a significant performance improvement.

this site is a bloody joke.

- 1) You're a cunt. As always.
- 2) You never give actual help or advice. Just abuse because you need serious mental help. You're seriously depressed or somewhat fucked in the head. Sorry you have to experience that.
- 3) I use StringBuilder like that based advice given me from [this](#) post. I was misinformed. It would've been cool if you could of--you know, elaborated constructively on when or when not to use it.
- 4) This site is a joke because people below your level of knowledge aren't at your level? Jesus Christ, you got a fucked up worldview.
- 5) Thanks for some knowledge. :)

<https://blog.codinghorror.com/the-sad-tragedy-of-micro-optimization-theater/>

Extra

Ninja - January 10, 2017, 7:25 am

The SHA1 object can be put in a using statement since it can also be disposed. And why make this a class instead of just a static method? Inb4 people bash me for saying the word static.

(Jan 9, 2017 06:59 PM)Malum™ Wrote: [\[CITE\]](#) 1) You're a cunt. As always.

2) You never give actual help or advice. Just abuse because you need serious mental help. You're seriously depressed or somewhat fucked in the head. Sorry you have to experience that.

3) I use StringBuilder like that based advice given me from [this](#) post. I was misinformed. It would've been cool if you could of--you know, elaborated constructively on when or when not to use it.

4) This site is a joke because people below your level of knowledge aren't at your level? Jesus Christ, you got a fucked up worldview.

5) Thanks for some knowledge. :)

<https://blog.codinghorror.com/the-sad-tragedy-of-micro-optimization-theater/>

Extra

To be fair minimal ms differences can be blamed on other background tasks the testing environment is running, but I agree, using a StringBuilder makes no difference.

Malum™ - January 10, 2017, 11:20 pm

(Jan 10, 2017 08:25 AM)Ninja Wrote: [\[CITE\]](#) The SHA1 object can be put in a using statement since it can also be disposed. And why make this a class instead of just a static method? Inb4 people bash me for saying the word static.

To be fair minimal ms differences can be blamed on other background tasks the testing environment is running, but I agree, using a StringBuilder makes no difference.

Thanks for the info on the SHA1! Honestly haven't converted it to static because I've just been lazy with it. You're right. It doesn't make sense to not be static. I might as well change it now while I'm motivated. Appreciate the agreement with string theories.

aneeshkollam5 - January 10, 2017, 11:46 pm

One important problem that i faced was when plug any external hdd or any usb device, hardwareid generated will change

I solved this issue by changing the code to

PHP Code

```
<?php
```

```
For Each wmi_HD As ManagementObject In searcher.[Get]()
```

```
    type = wmi_HD("InterfaceType").ToString()
```

```
    If type.ToUpper = "IDE" Or type.ToUpper = "SATA" Then return wmi_HD("SerialNumber").ToString
```

```
Next
```

any better alternative, or solution for getting only primary hdd serial number ?

Malum™ - January 11, 2017, 12:15 am

(Jan 11, 2017 12:46 AM)aneeshkollam5 Wrote: [\[CITE\]](#) One important problem that i faced was when plug any external hdd or any usb device, hardwareid generated will change

I solved this issue by changing the code to

PHP Code

```
<?php
```

```
For Each wmi_HD As ManagementObject In searcher.[Get]()
```

```
    type = wmi_HD("InterfaceType").ToString()
```

```
    If type.ToUpper = "IDE" Or type.ToUpper = "SATA" Then return wmi_HD("SerialNumber").ToString
```

```
Next
```

any better alternative, or solution for getting only primary hdd serial number ?

While looking around and researching it seems quite difficult to detect whether or not a USB device is an external HDD or will affect the HDD (Win32_DiskDrive) property as easily as the code was before. I'll have to look into it more. Using the code you gave gives me a completely different HWID and I have not changed any hardware. I greatly appreciate the information though! I'll look more into it.

ignorehf - January 11, 2017, 6:02 am

Your VB.NET code is broken.

Change this

```
Code For i As Integer = 0 to properties.Length
```

```
    sBuilder.Append(moObject.Properties(properties(i)))
```

```
Next
```

to this

```
Code For i As Integer = 0 To properties.Length - 1
```

```
    sBuilder.Append(mbObject.Properties(properties(i)))
```

```
Next
```

I don't know for others, but on my system HDD serial is not consistent using WMI. If you need consistency use pinvoke.