عنوان: **نحوه‌ی استخراج شماره سریال سخت افزار برای تولید یک قفل سخت افزاری**
نویسنده: احمد نواصری
تاریخ: ۱۳۹۴/۰۳/۲۷ ۱۰:۴۰
آدرس: www.dotnettips.info
گروه‌ها: C#, Security, WMI

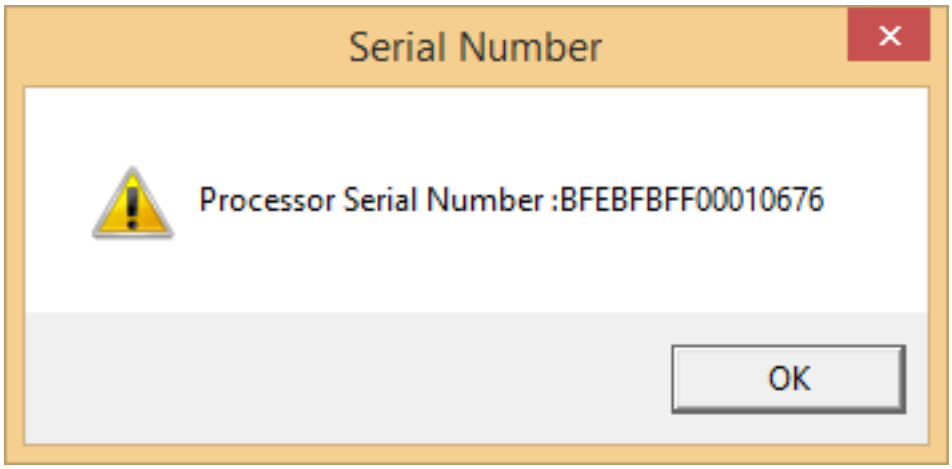
شما در حال نوشتن یک نرم افزار هستید و برای این نرم افزار ممکن است ماه‌ها وقت صرف کرده باشید؛ پس باید به دنبال راهی باشید که بتوانید از آن محافظت کنید. راه‌های متعددی برای Trial کردن نرم افزار وجود دارند که یکی از این راه‌ها استفاده از سریال سخت افزارهای کامپیوتر کاربر است. همانطور که می‌دانید هر سخت افزار یک شماره‌ی سریال مخصوص خودش را دارد و بدین طریق میتوان یک شماره سریال منحصر به فرد را تولید کرد. ما در این مقاله برای بدست آوردن کلیه‌ی مشخصات سخت افزار یک کامپیوتر از کلاس ManagementObjectSearcher در فضای نام System.Management استفاده کرده‌ایم.

```
using System.Management;
using System.Windows.Forms;
namespace HardwareSerialNumber
{
    class Program
    {
        static void Main()
        {
            string serialNumber = string.Empty;
            ManagementObjectSearcher searcher = new ManagementObjectSearcher("root\\CIMV2", "SELECT *
FROM Win32_Processor");
            foreach (var o in searcher.Get())
            {
                var query = (ManagementObject)o;

                serialNumber = serialNumber + query["ProcessorId"];
            }
            MessageBox.Show(string.Format("Processor Serial Number :{0}", serialNumber),"Serial
Number",MessageBoxButtons.OK,MessageBoxIcon.Exclamation);

        }
    }
}
```

خوب. حالا به بررسی کدهای بالا می‌پردازیم. در خط 10 یک شیء از کلاس MangementObjectSearcher ایجاد کرده‌ایم . سازنده‌ی این کلاس 2 پارامتر را می‌پذیرد. اولین پارامتر یک رشته می‌باشد root\\CIMV2 . CIMV2 فضای نام پیش فرض مخصوص کوئری‌های WMI می‌باشد و پارامتر دوم کوئری WMI است. عبارت Win32_Processor اشاره به کلاس Win32_processor دارد. این کلاس بیانگر یک کلاس مدیریتی از نوع CIM (مراجعه کنید به اینجا و اینجا) است. در خط 11 با استفاده از متد Get، تمامی اطلاعات مربوط به Processor را بارگذاری می‌کنیم و سپس فیلد ProcessorId را در یک رشته قرار می‌دهیم. نکته‌ی دیگر اینکه ارجاعی را به فایل System.Managment.Dll به پروژه‌ی خود اضافه کنید.
نتیجه‌ی کد بالا :



در این آدرس یک کتابخانه که شامل تمامی مثالها می‌باشد قابل در یافت است.

نظرات خوانندگان

نویسنده: فرهاد شربیانی
تاریخ: ۱۳۹۴/۰۳/۲۷ ۱۲:۴۸

می توان از این کلاس هم که در واقع بسط داده شده‌ی روش مذکور است استفاده نمود:

```
public class FingerPrint
    {
        private static string fingerPrint = string.Empty;
        public static string Value()
        {
            if (string.IsNullOrEmpty(fingerPrint))
            {
                fingerPrint = GetHash("CPU >> " + cpuId() + "\nBIOS >> " +
            biosId() + "\nBASE >> " + baseId()
                    +"\nDISK >> "+ diskId() + "\nVIDEO >> " +
                    videoId() +"\nMAC >> "+ macId());
            }
            return fingerPrint;
        }
        private static string GetHash(string s)
        {
            MD5 sec = new MD5CryptoServiceProvider();
            ASCIIEncoding enc = new ASCIIEncoding();
            byte[] bt = enc.GetBytes(s);
            return GetHexString(sec.ComputeHash(bt));
        }
        private static string GetHexString(byte[] bt)
        {
            string s = string.Empty;
            for (int i = 0; i < bt.Length; i++)
            {
                byte b = bt[i];
                int n, n1, n2;
                n = (int)b;
                n1 = n & 15;
                n2 = (n >> 4) & 15;
                if (n2 > 9)
                    s += ((char)(n2 - 10 + (int)'A')).ToString();
                else
                    s += n2.ToString();
                if (n1 > 9)
                    s += ((char)(n1 - 10 + (int)'A')).ToString();
                else
                    s += n1.ToString();
                if ((i + 1) != bt.Length && (i + 1) % 2 == 0) s += "-";
            }
            return s;
        }


        #region Original Device ID Getting Code
        //Return a hardware identifier
        private static string identifier
        (string wmiClass, string wmiProperty, string wmiMustBeTrue)
        {
            string result = "";
            System.Management.ManagementClass mc =
        new System.Management.ManagementClass(wmiClass);
            System.Management.ManagementObjectCollection moc = mc.GetInstances();
            foreach (System.Management.ManagementObject mo in moc)
            {
                if (mo[wmiMustBeTrue].ToString() == "True")
                {
                    //Only get the first one
                    if (result == "")
                    {
                        try
                        {
                            result = mo[wmiProperty].ToString();
                            break;
                        }
                        catch
                        {
                        }
                    }
```

```
            }
        }
        return result;
    }
    //Return a hardware identifier
    private static string identifier(string wmiClass, string wmiProperty)
    {
        string result = "";
        System.Management.ManagementClass mc =
    new System.Management.ManagementClass(wmiClass);
        System.Management.ManagementObjectCollection moc = mc.GetInstances();
        foreach (System.Management.ManagementObject mo in moc)
        {
            //Only get the first one
            if (result == "")
            {
                try
                {
                    result = mo[wmiProperty].ToString();
                    break;
                }
                catch
                {
                }
            }
        }
        return result;
    }
    private static string cpuId()
    {
        //Uses first CPU identifier available in order of preference
        //Don't get all identifiers, as it is very time consuming
        string retVal = identifier("Win32_Processor", "UniqueId");
        if (retVal == "") //If no UniqueID, use ProcessorID
        {
            retVal = identifier("Win32_Processor", "ProcessorId");
            if (retVal == "") //If no ProcessorId, use Name
            {
                retVal = identifier("Win32_Processor", "Name");
                if (retVal == "") //If no Name, use Manufacturer
                {
                    retVal = identifier("Win32_Processor", "Manufacturer");
                }
                //Add clock speed for extra security
                retVal += identifier("Win32_Processor", "MaxClockSpeed");
            }
        }
        return retVal;
    }
    //BIOS Identifier
    private static string biosId()
    {
        return identifier("Win32_BIOS", "Manufacturer")
            + identifier("Win32_BIOS", "SMBIOSBIOSVersion")
        + identifier("Win32_BIOS", "IdentificationCode")
            + identifier("Win32_BIOS", "SerialNumber")
            + identifier("Win32_BIOS", "ReleaseDate")
            + identifier("Win32_BIOS", "Version");
    }
    //Main physical hard drive ID
    private static string diskId()
    {
        return identifier("Win32_DiskDrive", "Model")
        + identifier("Win32_DiskDrive", "Manufacturer")
        + identifier("Win32_DiskDrive", "Signature")
        + identifier("Win32_DiskDrive", "TotalHeads");
    }
    //Motherboard ID
    private static string baseId()
    {
        return identifier("Win32_BaseBoard", "Model")
        + identifier("Win32_BaseBoard", "Manufacturer")
        + identifier("Win32_BaseBoard", "Name")
        + identifier("Win32_BaseBoard", "SerialNumber");
    }
    //Primary video controller ID
    private static string videoId()
    {
        return identifier("Win32_VideoController", "DriverVersion")
        + identifier("Win32_VideoController", "Name");
    }
```

```
        //First enabled network card ID
        private static string macId()
        {
            return identifier("Win32_NetworkAdapterConfiguration",
                "MACAddress", "IPEnabled");
        }
        #endregion
    }
```

خروجی این کلاس Hexal است و به شکل زیر قابل استفاده می‌باشد:

```
/// Generates a 16 byte Unique Identification code of a computer
/// Example: 4876-8DB5-EE85-69D3-FE52-8CF7-395D-2EA9

var computerId = FingerPrint.Value();
```

نویسنده: علی پناهی
تاریخ: ۱۳۹۴/۰۳/۲۷ ۱۶:۵۸

در لینک زیر هم می‌توانید لیست کاملی از تمام سخت افزارها را مشاهده نمایید لینک

نویسنده: احمد نواصری
تاریخ: ۱۳۹۴/۰۴/۰۱ ۱۸:۱۰

تمامی کلاسها و متدهای عنوان شده در این مثال، در کتابخانه ذکر شده در آخر بحث موجود می‌باشد.