**用于加密获取CPU序列号、硬盘ID、网卡MAC地址**

Posted on 2013年04月17日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 203 次

首先在添加引用中选中System.Management

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **using** System.Management; |
| 02 | **using** System.Management.Instrumentation; |
| 03 | **private** **void** GetInfo() |
| 04 | { |
| 05 | **string** cpuInfo = "";*//cpu序列号* |
| 06 | ManagementClass cimobject = new ManagementClass("Win32\_Processor"); |
| 07 | ManagementObjectCollection moc = cimobject.GetInstances(); |
| 08 | **foreach**(ManagementObject mo **in** moc) |
| 09 | { |
| 10 | cpuInfo = mo.Properties["ProcessorId"].**Value**.ToString(); |
| 11 | Response.Write ("cpu序列号："+cpuInfo.ToString ()); |
| 12 | } |
| 13 | *//获取硬盘ID* |
| 14 | **String** HDid; |
| 15 | ManagementClass cimobject1 = new ManagementClass("Win32\_DiskDrive"); |
| 16 | ManagementObjectCollection moc1 = cimobject1.GetInstances(); |
| 17 | **foreach**(ManagementObject mo **in** moc1) |
| 18 | { |
| 19 | HDid = (**string**)mo.Properties["Model"].**Value**; |
| 20 | Response.Write ("硬盘序列号："+HDid.ToString ()); |
| 21 | } |
| 22 |  |
| 23 | *//获取网卡硬件地址* |
| 24 |  |
| 25 |  |
| 26 | ManagementClass mc = new ManagementClass("Win32\_NetworkAdapterConfiguration"); |
| 27 | ManagementObjectCollection moc2 = mc.GetInstances(); |
| 28 | **foreach**(ManagementObject mo **in** moc2) |
| 29 | { |
| 30 | **if**((**bool**)mo["IPEnabled"] == **true**) |
| 31 | Response.Write("MAC address**\t**{0}"+mo["MacAddress"].ToString()); |
| 32 | mo.Dispose(); |
| 33 | } |
| 34 | } |
| 35 |  |
| 36 | **public** **static** **float** GetCPUPersent() |
| 37 | { |
| 38 | **float** cpuload = 0; |
| 39 | **const** **string** categoryname = "processor"; |
| 40 | **const** **string** countername = "% processor time"; |
| 41 | **const** **string** instancename = "\_total"; |
| 42 | PerformanceCounter pc = new PerformanceCounter(categoryname, countername, instancename); |
| 43 | **int** i = 10; |
| 44 | **while** (i > 0) |
| 45 | { |
| 46 | Thread.Sleep(1000); *// wait for 1 second* |
| 47 | cpuload = pc.NextValue(); |
| 48 | **if** (cpuload > 0) |
| 49 | { |
| 50 | **break**; |
| 51 | } |
| 52 | i--; |
| 53 | } |
| 54 | **return** cpuload; |
| 55 | } |
| 56 | **public** **static** **void** GetDiskSpace(**string** path, **out** **long** DiskAll, **out** **long** DiskActive) |
| 57 | { |
| 58 | DiskAll = 0; |
| 59 | DiskActive = 0; |
| 60 | **long** a, b, c; |
| 61 | **int** aaa = GetDiskFreeSpaceEx(path, **out** a, **out** b, **out** c); |
| 62 | DiskAll = (**long**)(b / 1024 / 1024); |
| 63 | DiskActive = (**long**)(a / 1024 / 1024); |
| 64 | } |
| 65 | **public** **static** **void** GetMemoryInfo(**out** **uint** MemoryAll, **out** **uint** MemoryUsed) |
| 66 | { |
| 67 | MemoryAll = 0; |
| 68 | MemoryUsed = 0; |
| 69 | MEMORY\_INFO MemInfo = new MEMORY\_INFO(); |
| 70 | GlobalMemoryStatus(**ref** MemInfo); |
| 71 | MemoryAll = MemInfo.dwTotalPhys / 1024 / 1024; |
| 72 | MemoryUsed = (MemInfo.dwTotalPhys - MemInfo.dwAvailPhys) / 1024 / 1024; |
| 73 | } |
| 74 |  |
| 75 |  |
| 76 |  |
| 77 | [DllImport("kernel32")] |
| 78 | **public** **static** **extern** **void** GlobalMemoryStatus(**ref** MEMORY\_INFO meminfo); |
| 79 | *//定义内存的信息结构* |
| 80 | [StructLayout(LayoutKind.Sequential)] |
| 81 | **public** **struct** MEMORY\_INFO |
| 82 | { |
| 83 | **public** **uint** dwLength; |
| 84 | **public** **uint** dwMemoryLoad; |
| 85 | **public** **uint** dwTotalPhys; |
| 86 | **public** **uint** dwAvailPhys; |
| 87 | **public** **uint** dwTotalPageFile; |
| 88 | **public** **uint** dwAvailPageFile; |
| 89 | **public** **uint** dwTotalVirtual; |
| 90 | **public** **uint** dwAvailVirtual; |
| 91 | } |
| 92 | [DllImport("kernel32.dll", EntryPoint = "GetDiskFreeSpaceExA")] |
| 93 | **public** **static** **extern** **int** GetDiskFreeSpaceEx(**string** lpRootPathName, **out** **long** lpFreeBytesAvailable, **out** **long** lpTotalNumberOfBytes, **out** **long** lpTotalNumberOfFreeBytes); |