**c#获取常用硬件信息**

c#中如何获取本机用户名、MAC地址、IP地址、硬盘ID、CPU序列号、系统名称、物理内存

首先需要引入命名空间：

using System.Management;

/// 操作系统的登录用户名：

string GetUserName()

{

try

{

string st = "";

ManagementClass mc = new ManagementClass("Win32\_ComputerSystem");

ManagementObjectCollection moc = mc.GetInstances();

foreach (ManagementObject mo in moc)

{

st = mo["UserName"].ToString();

}

moc = null;

mc = null;

return st;

}

catch

{

return "unknow";

}

finally

{ }

}

//获取MAC地址：

string GetMacAddress()

{

try

{

//获取网卡硬件地址

string mac = "";

ManagementClass mc = new ManagementClass("Win32\_NetworkAdapterConfiguration");

ManagementObjectCollection moc = mc.GetInstances();

foreach (ManagementObject mo in moc)

{

if ((bool)mo["IPEnabled"] == true)

{

mac = mo["MacAddress"].ToString();

break;

}

}

moc = null;

mc = null;

return mac;

}

catch

{

return "unknow";

}

finally

{ }

}

[复制代码](javascript:void(0);)

//获取IP地址：

[复制代码](javascript:void(0);)

/// <summary>

/// 获取客户端内网IP地址

/// </summary>

/// <returns></returns>

private static string getClientLocalIPAddress()

{

string localIP=null;

try

{

IPHostEntry ipHost = System.Net.Dns.GetHostEntry(Dns.GetHostName());// Dns.Resolve(Dns.GetHostName()); ;

IPAddress ipaddress = ipHost.AddressList[0];

localIP = ipaddress.ToString();

return "内网IP地址："+localIP;

}

catch

{

return "内网IP地址：unknown";

}

finally

{ }

}

/// <summary>

/// 获得客户端外网IP地址

/// </summary>

/// <returns>IP地址</returns>

public static string getClientInternetIPAddress()

{

string internetAddress = "";

try

{

using (WebClient webClient = new WebClient())

{

internetAddress = webClient.DownloadString("http://www.coridc.com/ip");//从外部网页获得IP地址

//判断IP是否合法

if (!System.Text.RegularExpressions.Regex.IsMatch(internetAddress, "[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}"))

{

internetAddress = webClient.DownloadString("http://fw.qq.com/ipaddress");//从腾讯提供的API中获得IP地址

}

}

return "外网IP地址：" + internetAddress;

}

catch

{

return "外网IP地址：unknown";

}

finally

{ }

}

//获取硬盘ID：

string GetDiskID()

{

try

{

String HDid = "";

ManagementClass mc = new ManagementClass("Win32\_DiskDrive");

ManagementObjectCollection moc = mc.GetInstances();

foreach (ManagementObject mo in moc)

{

//HDid = (string)mo.Properties["Model"].ToString();

HDid = (String)mo.Properties["Model"].Value.ToString();

}

moc = null;

mc = null;

return HDid;

}

catch

{

return "unknow";

}

finally

{ }

}

//获取CPUid：

string GetCpuID()

{

try

{

//获取CPU序列号代码

string cpuInfo = "";//cpu序列号

ManagementClass mc = new ManagementClass("Win32\_Processor");

ManagementObjectCollection moc = mc.GetInstances();

foreach (ManagementObject mo in moc)

{

cpuInfo = mo.Properties["ProcessorId"].Value.ToString();

}

moc = null;

mc = null;

return cpuInfo;

}

catch

{

return "unknow";

}

finally

{ }

}

//系统名称：

string GetSystemType()

{

try

{

string st = "";

ManagementClass mc = new ManagementClass("Win32\_ComputerSystem");

ManagementObjectCollection moc = mc.GetInstances();

foreach (ManagementObject mo in moc)

{

st = mo["SystemType"].ToString();

}

moc = null;

mc = null;

return st;

}

catch

{

return "unknow";

}

finally

{ }

}

//获取物理内存：

string GetTotalPhysicalMemory()

{

try

{

string st = "";

ManagementClass mc = new ManagementClass("Win32\_ComputerSystem");

ManagementObjectCollection moc = mc.GetInstances();

foreach (ManagementObject mo in moc)

{

st = mo["TotalPhysicalMemory"].ToString();

}

moc = null;

mc = null;

return st;

}

catch

{

return "unknow";

}

finally

{ }

}