**[C#程序调用cmd执行命令](http://www.cnblogs.com/babycool/p/3570648.html)**

来源: http://www.cnblogs.com/babycool/

对于C#通过程序来调用cmd命令的操作，网上有很多类似的文章，但很多都不行，竟是漫天的拷贝。我自己测试整理了一下。

代码：

string str = Console.ReadLine();

System.Diagnostics.Process p = new System.Diagnostics.Process();

p.StartInfo.FileName = "cmd.exe";

p.StartInfo.UseShellExecute = false; //是否使用操作系统shell启动

p.StartInfo.RedirectStandardInput = true;//接受来自调用程序的输入信息

p.StartInfo.RedirectStandardOutput = true;//由调用程序获取输出信息

p.StartInfo.RedirectStandardError = true;//重定向标准错误输出

p.StartInfo.CreateNoWindow = true;//不显示程序窗口

p.Start();//启动程序

//向cmd窗口发送输入信息

p.StandardInput.WriteLine(str + "&exit");

p.StandardInput.AutoFlush = true;

//p.StandardInput.WriteLine("exit");

//向标准输入写入要执行的命令。这里使用&是批处理命令的符号，表示前面一个命令不管是否执行成功都执行后面(exit)命令，如果不执行exit命令，后面调用ReadToEnd()方法会假死

//同类的符号还有&&和||前者表示必须前一个命令执行成功才会执行后面的命令，后者表示必须前一个命令执行失败才会执行后面的命令

//获取cmd窗口的输出信息

string output = p.StandardOutput.ReadToEnd();

//StreamReader reader = p.StandardOutput;

//string line=reader.ReadLine();

//while (!reader.EndOfStream)

//{

// str += line + " ";

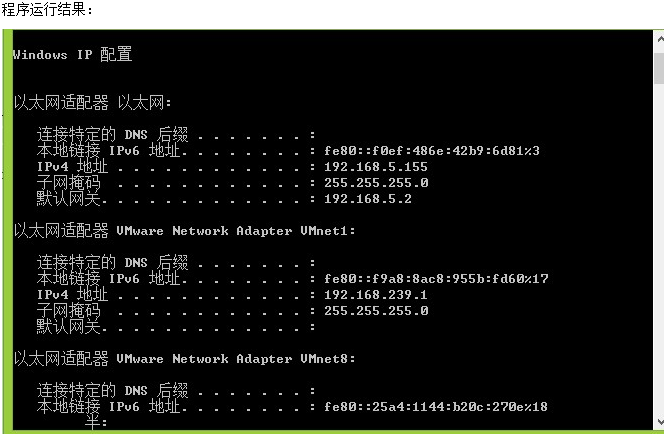
// line = reader.ReadLine();

//}

p.WaitForExit();//等待程序执行完退出进程

p.Close();

Console.WriteLine(output);



需要提醒注意的一个地方就是：在前面的命令执行完成后，要加exit命令，否则后面调用ReadtoEnd()命令会假死。

我在之前测试的时候没有加exit命令，输入其他命令后窗口就假死了，也没有输出内容。

2014-7-28 新增：

另一种C#调用cmd命令的方法，不过这种方法在执行时会“闪一下” 黑窗口，各位在使用时可以按喜好来调用。

/// <summary>

/// 运行cmd命令

/// 会显示命令窗口

/// </summary>

/// <param name="cmdExe">指定应用程序的完整路径</param>

/// <param name="cmdStr">执行命令行参数</param>

static bool RunCmd(string cmdExe, string cmdStr)

{

bool result = false;

try

{

using (Process myPro = new Process())

{

//指定启动进程是调用的应用程序和命令行参数

ProcessStartInfo psi = new ProcessStartInfo(cmdExe, cmdStr);

myPro.StartInfo = psi;

myPro.Start();

myPro.WaitForExit();

result = true;

}

}

catch

{

}

return result;

}

/// <summary>

/// 运行cmd命令

/// 不显示命令窗口

/// </summary>

/// <param name="cmdExe">指定应用程序的完整路径</param>

/// <param name="cmdStr">执行命令行参数</param>

static bool RunCmd2(string cmdExe, string cmdStr)

{

bool result = false;

try

{

using (Process myPro = new Process())

{

myPro.StartInfo.FileName = "cmd.exe";

myPro.StartInfo.UseShellExecute = false;

myPro.StartInfo.RedirectStandardInput = true;

myPro.StartInfo.RedirectStandardOutput = true;

myPro.StartInfo.RedirectStandardError = true;

myPro.StartInfo.CreateNoWindow = true;

myPro.Start();

                    //如果调用程序路径中有空格时，cmd命令执行失败，可以用双引号括起来 ，在这里两个引号表示一个引号（转义）

string str = string.Format(@"""{0}"" {1} {2}", cmdExe, cmdStr, "&exit");

myPro.StandardInput.WriteLine(str);

myPro.StandardInput.AutoFlush = true;

myPro.WaitForExit();

result = true;

}

}

catch

{

}

return result;

}