# 代码备忘

## 最新

Xlib 最新在自动登录vpn中

## 配置文件

操作ini文件

setConfString

getConfString

# 获取变量名

Nameof (vs2015)

# 打开控制台

//控制台

[System.Runtime.InteropServices.DllImport("kernel32.dll", SetLastError = true)]

[return: System.Runtime.InteropServices.MarshalAs(System.Runtime.InteropServices.UnmanagedType.Bool)]

public static extern bool openConsole();

[System.Runtime.InteropServices.DllImport("Kernel32")]

public static extern void FreeConsole();

# 文本框自动滚动到底部刷新

//使滚到到底部

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

richTextBox1.SelectionStart = richTextBox1.Text.Length;

richTextBox1.ScrollToCaret();

}

# Listviewe

View属性选details才是文字,默认是大图标模式

用时候要把属性columns里面先设置列明

双击事件

private void listView1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

ListViewHitTestInfo info = this.listView1.HitTest(e.X, e.Y);

if (info.Item != null)

{

MessageBox.Show(info.Item.Text);

}

}

# 网页操作webbrowser

不要放到多线程里,出错,不支持,就在主线程里就行了

但是可以放到自带的timer控件中去

//捕获控件的错误,再文档完成事件中添加就行了

 this.WebBrowser.Document.Window.Error += new HtmlElementErrorEventHandler(Window\_Error);

//对错误进行处理

 void Window\_Error(object sender, HtmlElementErrorEventArgs e)

        {

             // 自己的处理代码

            e.Handled = true;

        }

————————————————

版权声明：本文为CSDN博主「kingmax54212008」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/kingmax54212008/article/details/39559289

# 连接mysql数据库

Visual Studio,在 项目(右键)-管理NuGet程序包(N)  然后在浏览里面搜索MySql.Data并进行安装。安装6.7.9,不然最新版不支持net4

# 滚轮事件

属性窗口里是找不到的，但是可以手动添加代码

this.splitContainer.Panel1.MouseWheel += new System.Windows.Forms.MouseEventHandler(this.OnMouseWheel);

仅仅这样还不够，你会发现鼠标键盘还是不相应。

还需要在MouseDown事件里添加如下代码：

this.splitContainer.Panel1.Focus();

让控件获取焦点。

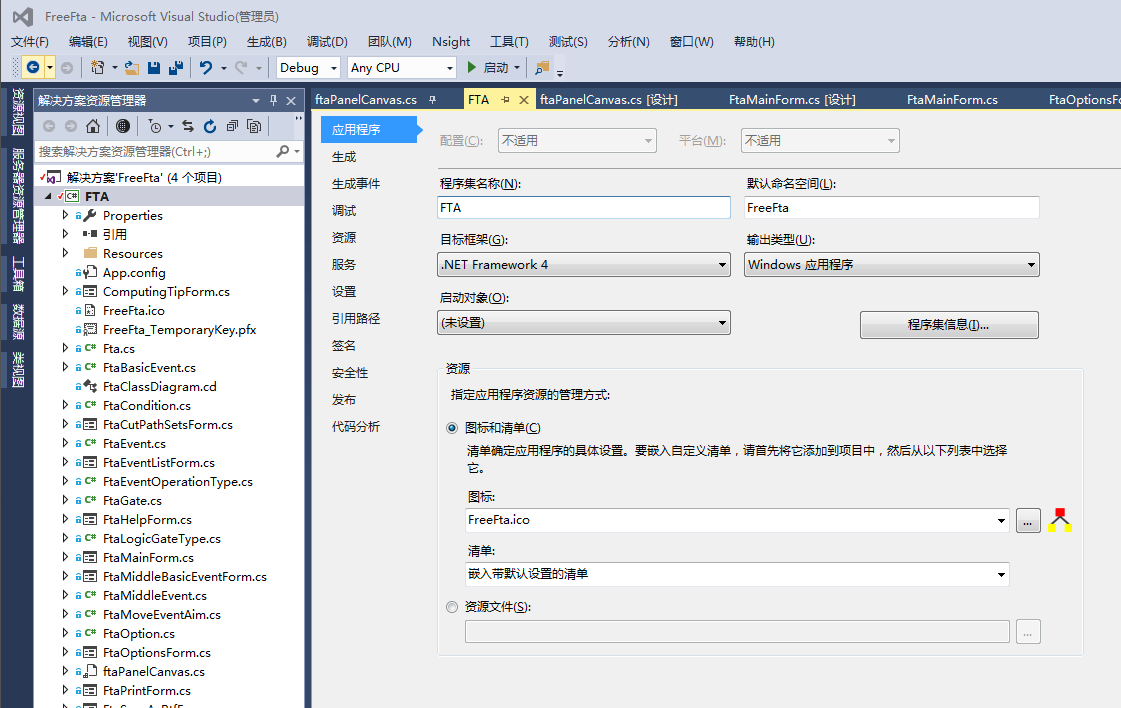
# 转换到 COFF 期间失败: 文件无效或损坏

方案一：（这个方法比较好，在用qt运行时出现问题也能解决）

复制 C:\Windows\winsxs\x86\_netfx-cvtres\_for\_vc\_and\_vb\_b03f5f7f11d50a3a\_6.1.7601.17514\_none\_ba1c770af0b2031b 目录下的 cvtres.exe 到C:\Program Files\Microsoft Visual Studio 10.0\VC\bin 目录下，直接覆盖替换。

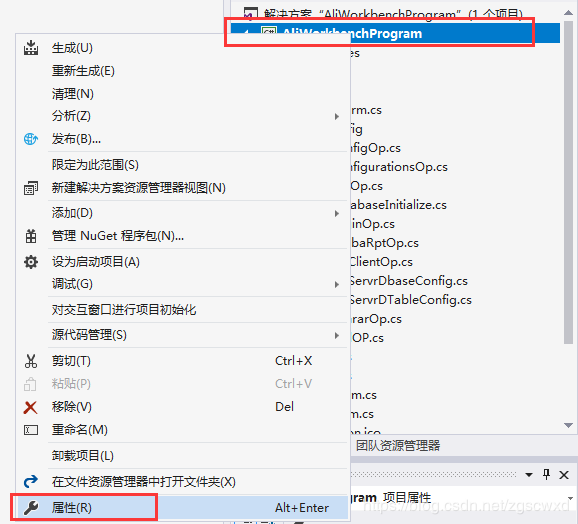
或C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin

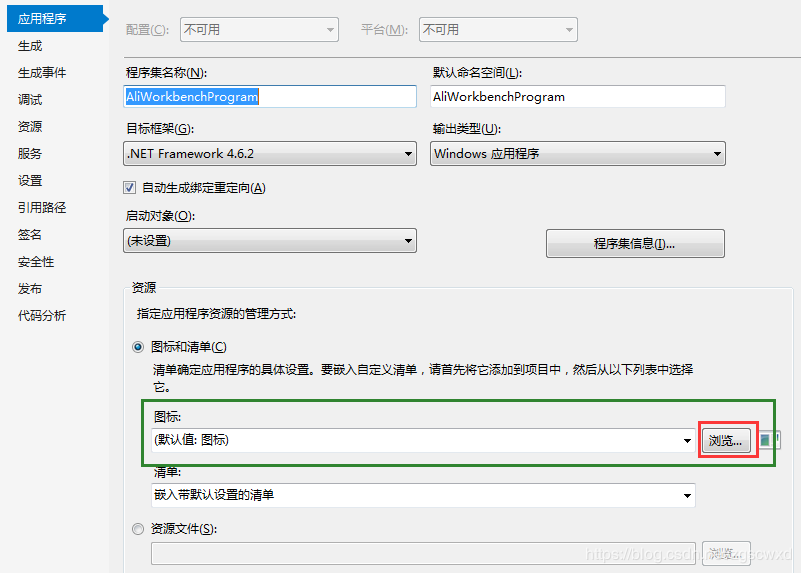
# 应用名称



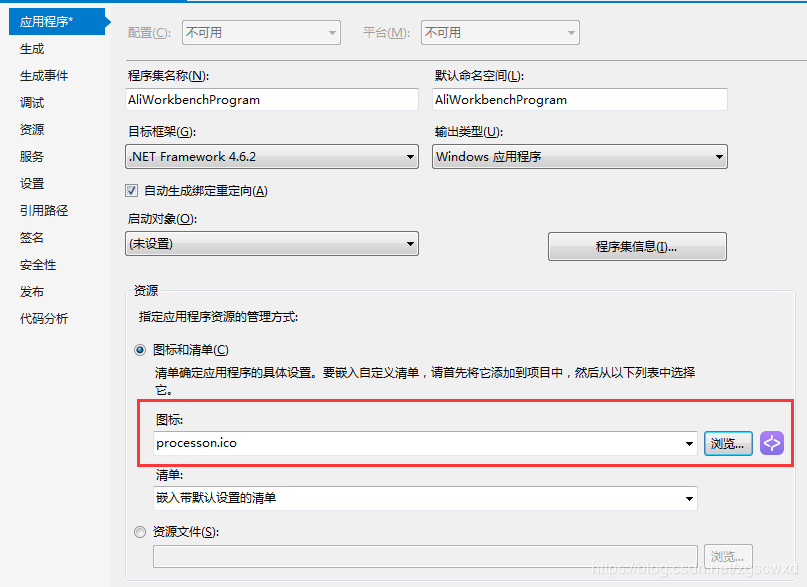
# 应用程序图标

下面介绍程序外部图标的修改方法：右击项目名称，在下拉列表中选择“属性”，如下图所示:





在弹出的窗口中单击图标右侧的“…”按钮，而后会弹出图标选择对话框，选择相应的图标后单击“打开”按钮，接着会发现图标已经修改成功,如下图所示:



此时，按下“F5”运行程序，同时也是为了保存修改后的图标，运行界面如下，可以看出，程序内部的图标已经发生改变;

# 窗口

## 透明

要实现C# WinForm中的控件与背景的透明，可以通过设置控件的BackColor属性为Transparent，同时设置其父控件。因为在C#中，控件的透明指对父窗体透明。  
如果不设置Parent属性，那么控件将只对Form透明，显示的时候都会把Form的背景色（默认为Control）重刷一遍作为自己的背景。  
在控件比较多的情况下，可以使用Panel控件，将某一组的控件都放到Panel中，然后只对此panel设置背景透明即可，则其他控件都跟着实现了背景透明。  
如，现有一PictureBox控件，十多个Label以及Button，那么只将这些Label和Button放入Panel中。同时在Form\_Load事件中加入如下代码即可实现背景透明：  
   this.picturebox1.SendToBack();//将背景图片放到最下面  
   this.panel1.BackColor = Color.Transparent;//将Panel设为透明  
   this.panel1.Parent = this.picturebox1;//将panel父控件设为背景图片控件  
   this.panel1.BringToFront();//将panel放在前面  
以上代码即可实现所有的控件都对PictueBox背景图片透明

## 更改控件顺序

右键置于底层

## 非模态窗口且新的窗口始终显示在前边

FormNew fn = new FormNew();

//fn.ShowDialog();

fn.TopMost = true;

fn.Show();

## 刷新

fn.Invalidate();

## 2. 非模态窗口方式(可以跟其他界面自由切换，而且不阻塞代码)

   Show()方法启动的窗口是非模态窗口，可以跟其他界面自由切换，

   其后面的代码也会立刻执行，如：

    Form1 form1 = new Form1();

    form1.Show();

    MessageBos.Show("这里会马上显示");

## 3. 模态窗口(必须关闭了该窗口，后面的代码才会执行，并且不能跟其他界面自由切换)

       ShowDialog()打开的窗口为模态窗口(只是个名字而已)，

       此类窗口不能和别的窗口自由切换，而且其后面的代码必须等关闭该窗口才会执行（就像被阻塞在那里了）

       如：

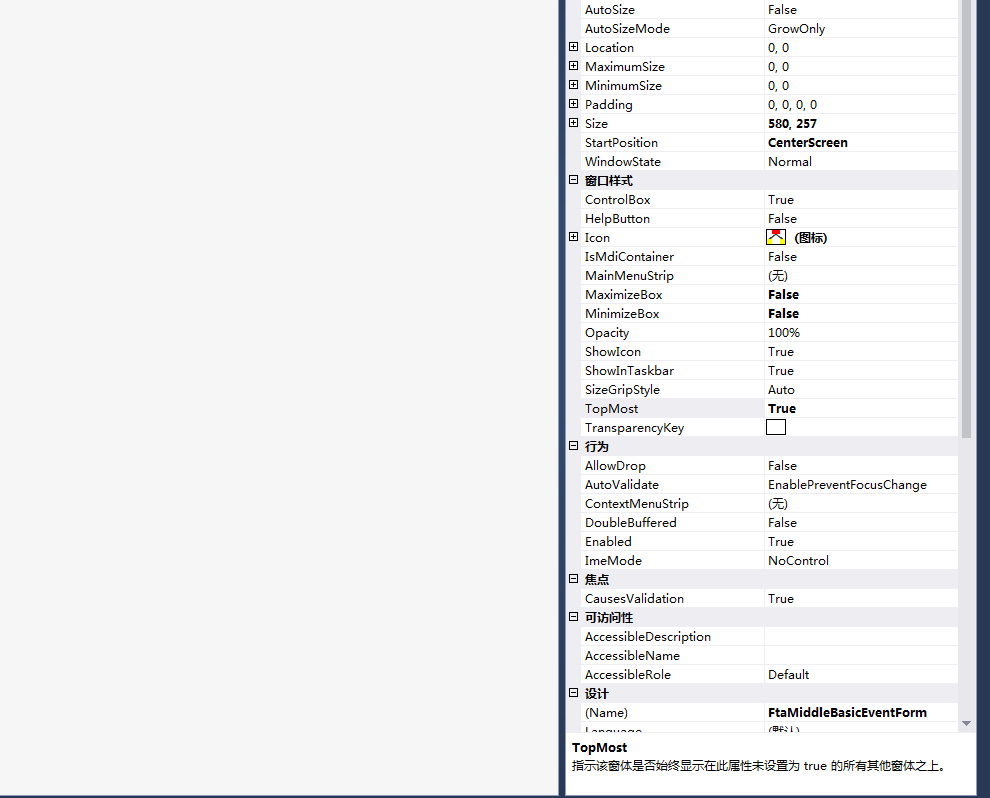
         Form1 form1 = new Form1();

         form1.ShowDialog();

         MessageBos.Show("必须关闭了form1, 这里才会显示");

## 窗口置顶

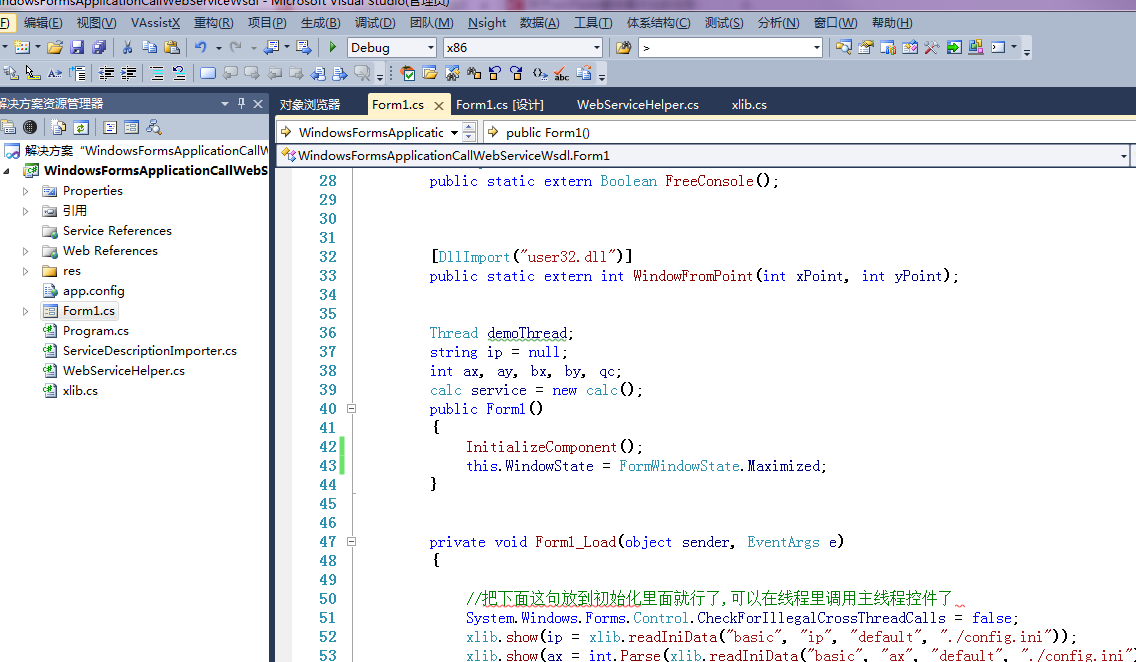
Topmost



# 图像

要先弄一个panel再在里面弄picturebox,panel中药选autoscroll,anchor选topleft还有图片框也要一样

# 打开即最大化



# Socket

## 连接域名

前几天用了花生壳 免费域名

用域名连接 感觉不错 重点是这个内网穿透 要不只能天天自己单机测试 要不就买个与服务器

之前大学没毕业某云服务器便宜 买了两个月 把我的毕设挂上了 Tomcat做服务器 确实好使

但是现在已经毕业了 还不想花钱 就拿这个很不错

 然后用Photon做服务器 客户端是uLua自带的network 进行数据传输

确实在本机用域名（外网正要试试）连上了

但是今天我打算用个最简单的测试版本跟朋友用外网测试的时候（这里两端用的都是c#  socket）却发现问题：

 IPAddress ipAddress = IPAddress.Parse("www.xxxxxx.xxx");

连接的时候报错 后来百度 发现域名应该用下面代码解析：

IPHostEntry hostInfo = Dns.GetHostEntry("www.xxxxxx.xxx");  
IPAddress ipAddress = hostInfo.AddressList[0];   
 这样 在本机用域名连接就好使了

一会试试外网连接好使不

。。。。。。

第二天了 修改一下 确实好使了 这样以后就可以拿本机当服务器 然后让朋友们都测试就可以了

## server

usingSystem**;**

usingSystem.Collections.Generic**;**

usingSystem.ComponentModel**;**

usingSystem.Data**;**

usingSystem.Linq**;**

usingSystem.Text**;**

usingSystem.Runtime.InteropServices**;**

usingSystem.Threading**;**

usingSystem.Net**;**

usingSystem.Net.Sockets**;**

namespaceConsoleApplication1

**{**

classProgram

**{**

//记?录?通ª¡§信?用®?的Ì?Socket

staticDictionary<string**,** Socket>dic=newDictionary<string**,** Socket>**();**

// private Socket client;

staticvoidAcceptInfo**(**objecto**)**

**{**

Socketsocket=oasSocket**;**

while **(**true**)**

**{**

//通ª¡§信?用®?socket

try

**{**

//创ä¡ä建¡§通ª¡§信?用®?的Ì?Socket

SockettSocket=socket.Accept**();**

stringpoint=tSocket.RemoteEndPoint.ToString**();**

//IPEndPoint endPoint = (IPEndPoint)client.RemoteEndPoint;

//string me = Dns.GetHostName();//得Ì?到Ì?本À?机¨²名?称?

//MessageBox.Show(me);

Console.WriteLine**(**point+"连¢?接¨®成¨¦功|！ê?"**);**

dic.Add**(**point**,** tSocket**);**

//接¨®收º?消?息¡é

Threadth=newThread**(**ReceiveMsg**);**

th.IsBackground=true**;**

th.Start**(**tSocket**);**

**}**

catch **(**Exceptionex**)**

**{**

Console.WriteLine**(**ex.Message**);**

break**;**

**}**

**}**

**}**

//接¨®收º?消?息¡é

staticvoidReceiveMsg**(**objecto**)**

**{**

Socketclient=oasSocket**;**

while **(**true**)**

**{**

//接¨®收º?客¨ª户¡ì端?发¤¡é送¨ª过y来¤¡ä的Ì?数ºy据Y

try

**{**

//定¡§义°?byte数ºy组Á¨¦存ä?放¤?从ä¨®客¨ª户¡ì端?接¨®收º?过y来¤¡ä的Ì?数ºy据Y

byte**[]** buffer=newbyte**[**1024\*1024**];**

//将?接¨®收º?过y来¤¡ä的Ì?数ºy据Y放¤?到Ì?buffer中D，ê?并¡é返¤¦Ì回?实º¦Ì际¨º接¨®受º¨¹数ºy据Y的Ì?长¡è度¨¨

intn=client.Receive**(**buffer**);**

//将?字Á?节¨²转Áa换?成¨¦字Á?符¤?串ä?

stringwords=Encoding.UTF8.GetString**(**buffer**,** 0**,** n**);**

Console.WriteLine**(**client.RemoteEndPoint.ToString**()** +":"+words**);**

**}**

catch **(**Exceptionex**)**

**{**

Console.WriteLine**(**ex.Message**);**

break**;**

**}**

**}**

**}**

staticvoidMain**(**string**[]** args**)**

**{**

//ip地Ì?址¡¤

IPAddressip=IPAddress.Parse**(**"192.168.31.35"**);**

IPEndPointpoint=newIPEndPoint**(**ip**,**int.Parse**(**"4444"**));**

//创ä¡ä建¡§监¨¤听¬y用®?的Ì?Socket

/\*

\* AddressFamily.InterNetWork：êo使º1用®? IP4地Ì?址¡¤。¡êSocketType.Stream：êo支¡ì持?可¨¦靠?、¡é双?向¨°、¡é基¨´于®¨²连¢?接¨®的Ì?字Á?节¨²流¢¡Â，ê?而?不?重?复¡ä数ºy据Y。¡ê

\* 此ä?类¤¨¤型¨ª的Ì? Socket 与®?单Ì£¤个?对?方¤?主¡Â机¨²进?行D通ª¡§信?，ê?并¡é且¨°在¨²通ª¡§信?开a始º?之?前¡ã需¨¨要°a远?程¨¬主¡Â机¨²连¢?接¨®。¡ê

\* Stream 使º1用®?传ä?输º?控?制?协-议°¨¦ (Tcp) ProtocolType 和¨ª InterNetworkAddressFamily。¡êProtocolType.Tcp：êo使º1用®?传ä?输º?控?制?协-议°¨¦。¡ê

\*/

//使º1用®?IPv4地Ì?址¡¤，ê?流¢¡Â式º?socket方¤?式º?，ê?tcp协-议°¨¦传ä?递ÌY数ºy据Y

Socketsocket=newSocket**(**AddressFamily.InterNetwork**,**SocketType.Stream**,**ProtocolType.Tcp**);**

//创ä¡ä建¡§好?socket后¨®，ê?必À?须?告?诉?socket绑ã¨®定¡§的Ì?IP地Ì?址¡¤和¨ª端?口¨²号?。¡ê

//让¨?socket监¨¤听¬ypoint

try

**{**

//socket监¨¤听¬y哪?个?端?口¨²

socket.Bind**(**point**);**

//同ª?一°?个?时º¡À间?点Ì?过y来¤¡ä10个?客¨ª户¡ì端?，ê?排?队¨®

socket.Listen**(**10**);**

Console.WriteLine**(**"服¤t务?器¡Â开a始º?监¨¤听¬y"**);**

Threadthread=newThread**(**AcceptInfo**);**

thread.IsBackground=true**;**

thread.Start**(**socket**);**

**}**

catch **(**Exceptionex**)**

**{**

Console.WriteLine**(**ex.Message**);**

**}**

Console.ReadKey**();**

**}**

**}**

**}**

## client

Socketclient=newSocket**(**AddressFamily.InterNetwork**,** SocketType.Stream**,** ProtocolType.Tcp**);**

//接¨®收º?服¤t务?器¡Â的Ì?消?息¡é

voidSendMsg**()**

**{**

while **(**true**)**

**{**

try

**{**

//send message

stringsendStr="test"**;**

byte**[]** sendBytes=Encoding.ASCII.GetBytes**(**sendStr**);**

client.Send**(**sendBytes**);**

**}**

catch **(**Exceptionex**)**

**{**

System.Environment.Exit**(**0**);**

//MessageBox.Show(ex.Message);

//break;

**}**

Thread.Sleep**(**500**);**

**}**

**}**

privatevoidbutton1\_Click**(**objectsender**,** EventArgse**)**

**{**

//连¢?接¨®到Ì?的Ì?目?标À¨ºIP

IPAddressip=IPAddress.Parse**(**"192.168.31.35"**);**

//IPAddress ip = IPAddress.Any;

//连¢?接¨®到Ì?目?标À¨ºIP的Ì?哪?个?应®|用®?(端?口¨²号?！ê?)

IPEndPointpoint=newIPEndPoint**(**ip**,**int.Parse**(**"4444"**));**

try

**{**

//连¢?接¨®到Ì?服¤t务?器¡Â

client.Connect**(**point**);**

//MessageBox.Show("连¢?接¨®成¨¦功|! 服¤t务?器¡Â: " + client.RemoteEndPoint.ToString() + "客¨ª户¡ì端?: " + client.LocalEndPoint.ToString());

//连¢?接¨®成¨¦功|后¨®，ê?就¨ª可¨¦以°?接¨®收º?服¤t务?器¡Â发¤¡é送¨ª的Ì?信?息¡é了¢?

Threadth=newThread**(**SendMsg**);**

th.IsBackground=true**;**

th.Start**();**

**}**

catch **(**Exceptionex**)**

**{**

MessageBox.Show**(**ex.Message**);**

**}**

**}**

# 最新libs.xlib

在自动登陆vpn中

# 退出程序

1.this.Close();   只是关闭当前窗口，若不是主窗体的话，是无法退出程序的，另外若有托管线程（非主线程），也无法干净地退出；

2.Application.Exit();  强制所有消息中止，退出所有的窗体，但是若有托管线程（非主线程），也无法干净地退出；

3.Application.ExitThread(); 强制中止调用线程上的所有消息，同样面临其它线程无法正确退出的问题；

4.System.Environment.Exit(0);   这是最彻底的退出方式，不管什么线程都被强制退出，把程序结束的很干净。

System.Environment.Exit**(**0**);**

## Form

usingSystem**;**

usingSystem.Collections.Generic**;**

usingSystem.ComponentModel**;**

usingSystem.Data**;**

usingSystem.Drawing**;**

usingSystem.Linq**;**

usingSystem.Text**;**

usingSystem.Windows.Forms**;**

usingSystem.Runtime.InteropServices**;**

usingSystem.Threading**;**

usinglibs**;**

namespace自Á?动¡¥登Ì?陆?vpn

**{**

publicpartialclassForm1 **:** Form

**{**

//查¨¦找¨°窗ä¡ã体¬?

**[**DllImport**(**"User32.dll"**,** EntryPoint="FindWindow"**)]**

publicexternstaticIntPtrFindWindow**(**stringlpClassName**,** stringlpWindowName**);**

**[**DllImport**(**"User32.dll"**,** EntryPoint="FindWindowEx"**)]**

publicstaticexternIntPtrFindWindowEx**(**IntPtrhwndParent**,** IntPtrhwndChildAfter**,** stringlpClassName**,** stringlpWindowName**);**

**[**DllImport**(**"user32.dll"**,** CharSet=CharSet.Auto**,** SetLastError=false**)]**

staticexternIntPtrSendMessage**(**IntPtrhWnd**,** uintMsg**,** IntPtrwParam**,** stringlParam**);**

**[**DllImport**(**"user32.dll"**,** SetLastError=true**)]**

staticexternIntPtrGetWindow**(**IntPtrhWnd**,** uintuCmd**);**

**[**DllImport**(**"user32.dll"**,** CharSet=CharSet.Auto**)]**

staticexternintGetClassName**(**IntPtrhWnd**,** StringBuilderlpClassName**,** intnMaxCount**);**

ThreaddemoThread**;**

publicvoiddelay**(**intms**)**

**{**

DateTimecurrent=DateTime.Now**;**

while **(**current.AddMilliseconds**(**ms**)** >DateTime.Now**)**

**{**

Application.DoEvents**();**

**}**

return**;**

**}**

publicForm1**()**

**{**

InitializeComponent**();**

**}**

//通ª¡§过y索¡Â引°y查¨¦找¨°相¨¤应®|控?件t句?柄À¨²

staticIntPtrFindWindowByIndex**(**IntPtrhwndParent**,** intindex**)**

**{**

if **(**index==0**)**

**{**

returnhwndParent**;**

**}**

else

**{**

intct=0**;**

IntPtrresult=IntPtr.Zero**;**

do

**{**

result=FindWindowEx**(**hwndParent**,** result**,** null**,** null**);**

if **(**result!=IntPtr.Zero**)**

**{**

++ct**;**

**}**

**}** while **(**ct<index&&result!=IntPtr.Zero**);**

returnresult**;**

**}**

**}**

privateboolcloseProc**(**stringProcName**){**

boolresult=false**;**

System.Collections.ArrayListprocList=newSystem.Collections.ArrayList**();**

stringtempName=""**;**

foreach **(**System.Diagnostics.ProcessthisProcinSystem.Diagnostics.Process.GetProcesses**())**

**{**

tempName=thisProc.ProcessName**;**

procList.Add**(**tempName**);**

if **(**tempName==ProcName**)**

**{**

if **(**!thisProc.CloseMainWindow**())　?　?**

thisProc.Kill**();** //当Ì¡À发¤¡é送¨ª关?闭À?窗ä¡ã口¨²命¨¹令¢?无T效¡ì时º¡À强?行D结¨¢束º?进?程¨¬

result=true**;**

**}**

**}**

returnresult**;**

**}**

privatevoidThreadProcSafe**()**

**{**

//int icount = 0;//警¡¥报À¡§次ä?数ºy

while **(**true**)**

**{**

//closeProc("脚?本À?错ä¨ª误¨®");//关?闭À?脚?本À?错ä¨ª误¨®窗ä¡ã口¨²

IntPtrhscripterror=FindWindow**(**null**,** "脚?本À?错ä¨ª误¨®"**);**

if **(**hscripterror!=IntPtr.Zero**)**

SendMessage**(**hscripterror**,** 0x0010**,** IntPtr.Zero**,** null**);**

IntPtrmaindHwnd=FindWindow**(**null**,** "安ã2全¨?警¡¥报À¡§"**);** //安ã2全¨?警¡¥报À¡§要°a点Ì?是º?

if **(**maindHwnd!=IntPtr.Zero**)**

**{**

//MessageBox.Show("找¨°到Ì?了¢?！ê?");

IntPtrbuth=FindWindowEx**(**maindHwnd**,** IntPtr.Zero**,** "Button"**,** null**);**

//IntPtr buth = FindWindowByIndex(maindHwnd, 3);

if **(**buth!=IntPtr.Zero**)**

**{**

//点Ì?击¡Â确¨¡¤定¡§

SendMessage**(**buth**,** 0xF5**,** IntPtr.Zero**,** null**);**

**}**

**}**

//以°?下?是º?piranha登Ì?陆?的Ì?

IntPtrhsecure=FindWindow**(**null**,** "Windows 安ã2全¨?"**);**

if **(**hsecure!=IntPtr.Zero**)**

**{**

IntPtrh2=FindWindowEx**(**hsecure**,** IntPtr.Zero**,** "DirectUIHWND"**,** null**);**

//根¨´据Y索¡Â引°y,就¨ª是º?父?控?件t下?的Ì?第Ì¨²几?个?控?件t

IntPtrhuser=FindWindowByIndex**(**h2**,** 7**);**

IntPtrhuser2=FindWindowEx**(**huser**,** IntPtr.Zero**,** "Edit"**,** null**);**

IntPtrhpwd=FindWindowByIndex**(**h2**,** 8**);**

IntPtrhpwd2=FindWindowEx**(**hpwd**,** IntPtr.Zero**,** "Edit"**,** null**);**

IntPtrhbutok=FindWindowByIndex**(**h2**,** 3**);**

IntPtrhbutok2=FindWindowEx**(**hbutok**,** IntPtr.Zero**,** "Button"**,** "确¨¡¤定¡§"**);**

if **(**huser2!=IntPtr.Zero**)**

//输º?入¨?字Á?符¤?串ä?

SendMessage**(**huser2**,** 0x000C**,** IntPtr.Zero**,** "piranha"**);**

if **(**hpwd2!=IntPtr.Zero**)**

SendMessage**(**hpwd2**,** 0x000C**,** IntPtr.Zero**,** "piranha"**);**

if **(**hbutok!=IntPtr.Zero**)**

// //点Ì?击¡Â确¨¡¤定¡§

SendMessage**(**hbutok2**,** 0xF5**,** IntPtr.Zero**,** null**);**

**}**

Thread.Sleep**(**100**);**

**}**

**}**

privateboolWaitWebPageLoad**(**WebBrowserbrowser**)**

**{**

inti=0**;**

stringsUrl**;**

while **(**true**)**

**{**

delay**(**50**);** //系¦Ì统ª3延¨®迟¨´50毫¨¢秒?，ê?够?少¦¨´了¢?吧ã¨¦！ê?

if **(**browser.ReadyState==WebBrowserReadyState.Complete**)** //先¨¨判D断?是º?否¤?发¤¡é生¦¨²完ª¨º成¨¦事º?件t。¡ê

**{**

if **(**!browser.IsBusy**)** //再¨´判D断?是º?浏¡¥览¤¨¤器¡Â是º?否¤?繁¤¡À忙|

**{**

i=i+1**;**

if **(**i==2**)** //为a什º2么¡ä 是º?2呢?？ê?因°¨°为a每?次ä?加¨®载?frame完ª¨º成¨¦时º¡À就¨ª会¨¢置?IsBusy为afalse,未¡ä完ª¨º成¨¦就¨ª就¨ª置?IsBusy为afalse，ê?你?想?一°?想?，ê?加¨®载?一°?次ä?，ê?然¨?后¨®再¨´一°?次ä?，ê?再¨´一°?次ä?...... 最Á?后¨®一°?次ä?.......

**{**

sUrl=browser.Url.ToString**();**

if **(**sUrl.Contains**(**"res"**))** //这a是º?判D断?没?有®D网ª?络?的Ì?情¨¦况?下?

**{**

returnfalse**;**

**}**

else

**{**

returntrue**;**

**}**

**}**

continue**;**

**}**

i=0**;**

**}**

**}**

**}**

privatevoidForm1\_Load**(**objectsender**,** EventArgse**)**

**{**

this.demoThread=newThread**(**newThreadStart**(**this.ThreadProcSafe**));**

this.demoThread.Start**();**

//如¨?果?ping不?通ª¡§,则¨°要°a登Ì?陆?vpn

if **(**!xlib.ping**(**"172.16.12.1"**))**

**{**

//先¨¨关?闭À?原-来¤¡ä的Ì?ssl客¨ª户¡ì端?,不?然¨?启?动¡¥失º¡ì败ã¨¹

closeProc**(**"VPNManage"**);**

webBrowser1.Navigate**(**"http://vpn.ceode.ac.cn"**);**

WaitWebPageLoad**(**webBrowser1**);**

HtmlElementuser=webBrowser1.Document.All**[**"txtUsrName"**];**

HtmlElementpwd=webBrowser1.Document.All**[**"txtPassword"**];**

HtmlElementbut=webBrowser1.Document.All**[**"Submit"**];**

user.SetAttribute**(**"value"**,** "panjie"**);**

pwd.SetAttribute**(**"value"**,** "15810805214"**);**

but.InvokeMember**(**"click"**);**

**}**

**}**

privatevoidlogin\_cluster**(**WebBrowserbrowser**,**stringurl**)**

**{**

//browser.Navigate("about:blank");

browser.Navigate**(**url**);**

WaitWebPageLoad**(**browser**);**

//HtmlElement user = webBrowser2.Document.All["login"];//这a句?根¨´据Yid不?管¨¹用®?,可¨¦能¨¹有®D多¨¤个?

HtmlElementpwd=browser.Document.All**[**"password"**];**

HtmlElementbut=browser.Document.All**[**"submit"**];**

if **(**pwd==null**)**

**{**

//label1.Text = "出?现?未¡ä知a错ä¨ª误¨®,请?重?试º?!";

//MessageBox.Show(label1.Text);

return**;**

**}**

pwd.SetAttribute**(**"value"**,** "scfwq\_325"**);**

//user.SetAttribute("value", "root");

HtmlElementu=browser.Document.All.GetElementsByName**(**"login"**)[**0**];**//用®?这a句?才?管¨¹用®?

u.SetAttribute**(**"value"**,** "root"**);**

but.InvokeMember**(**"click"**);**

label1.Text=url**;**

**}**

privatevoidgfksjq\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_cluster**(**webBrowser2**,** "https://172.16.141.7:8084"**);**

//webBrowser2.Navigate("https://172.16.141.7:8084");

//WaitWebPageLoad(webBrowser2);

////HtmlElement user = webBrowser2.Document.All["login"];//这a句?根¨´据Yid不?管¨¹用®?,可¨¦能¨¹有®D多¨¤个?

//HtmlElement pwd = webBrowser2.Document.All["password"];

//HtmlElement but = webBrowser2.Document.All["submit"];

//pwd.SetAttribute("value", "scfwq\_325");

////user.SetAttribute("value", "root");

//HtmlElement u = webBrowser2.Document.All.GetElementsByName("login")[0];//用®?这a句?才?管¨¹用®?

//u.SetAttribute("value", "root");

//but.InvokeMember("click");

**}**

privatevoidbutton\_gfdtcjq\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_cluster**(**webBrowser2**,** "https://172.16.11.17:8084"**);**

**}**

privatevoidbutton\_gfmyjq\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_cluster**(**webBrowser2**,** "https://172.16.77.7:8084"**);**

**}**

privatevoidbutton\_gfsyjq\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_cluster**(**webBrowser2**,** "https://172.16.171.7:8084"**);**

**}**

privatevoidbutton\_z3myjq\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_cluster**(**webBrowser2**,** "https://172.16.80.10:8084"**);**

**}**

privatevoidbutton\_z3ksjq\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_cluster**(**webBrowser2**,** "https://172.16.143.10:8084"**);**

**}**

privatevoidlogin\_piranha**(**WebBrowserb**,**stringurl**)**

**{**

b.Navigate**(**url**);**

WaitWebPageLoad**(**b**);**

// 如¨?果?整?个?页°3面?就¨ª一°?个?表À¨ª单Ì£¤.

//想?提¬¨¢交?表À¨ª单Ì£¤就¨ª写¡ä:

//this.webBrowser2.Document.Forms[0].InvokeMember("submit");

//如¨?果?相¨¤应®|的Ì?只?有®D一°?个?subimt按ã¡ä钮£¤.

//想?点Ì?击¡Â提¬¨¢交?按ã¡ä钮£¤就¨ª写¡ä:

//根¨´据Y表À¨ª单Ì£¤查¨¦找¨°所¨´有®Dinput

HtmlElementCollectioninput=b.Document.GetElementsByTagName**(**"input"**);**

for **(**intii=0**;** ii<input.Count**;** ii++**)**

**{**

if **(**input**[**ii**]**.GetAttribute**(**"type"**)**.ToLower**()**.Equals**(**"submit"**))**

**{**

input**[**ii**]**.InvokeMember**(**"click"**);**

**}**

**}**

**}**

privatevoidbutton\_gfmyfzjh\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_piranha**(**webBrowser2**,** "http://172.16.77.101:3636/"**);**

**}**

privatevoidbutton\_gfsyfzjh\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_piranha**(**webBrowser2**,** "http://172.16.171.100:3636"**);**

**}**

privatevoidbutton\_gfksfzjh\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_piranha**(**webBrowser2**,** "http://172.16.141.3:3636"**);**

**}**

privatevoidbutton\_xddtcfzjh\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_piranha**(**webBrowser2**,** "http://172.16.12.7:3636"**);**

**}**

privatevoidbutton\_xdmycfzjh\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_piranha**(**webBrowser2**,** "http://172.16.78.100:3636"**);**

**}**

privatevoidbutton\_xdsyfzjh\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_piranha**(**webBrowser2**,** "http://172.16.172.100:3636"**);**

**}**

privatevoidbutton\_xdksfzjh\_Click**(**objectsender**,** EventArgse**)**

**{**

login\_piranha**(**webBrowser2**,** "http://172.16.142.100:3636"**);**

**}**

privatevoidForm1\_FormClosed**(**objectsender**,** FormClosedEventArgse**)**

**{**

demoThread.Abort**();**

**}**

**}**

**}**

## Xlib

usingSystem**;**

usingSystem.Collections.Generic**;**

usingSystem.ComponentModel**;**

usingSystem.Data**;**

usingSystem.Drawing**;**

usingSystem.Linq**;**

usingSystem.Text**;**

usingSystem.Windows.Forms**;**

usingSystem.Runtime.InteropServices**;**

usingSystem.Threading**;**

usingSystem.IO**;**

usingSystem.Net.NetworkInformation**;**

usingSystem.Data.OleDb**;**

//该?类¤¨¤是º?winform的Ì?类¤¨¤

namespacelibs

**{**

publicclassxlib

**{**

#region

publicconstintMK\_LBUTTON=0x0001**;**

publicconstintWM\_SYSCOMMAND=0x0112**;**

publicconstintSC\_MINIMIZE=0xF020**;**

publicconstintBM\_CLICK=0xF5**;**//点Ì?击¡Â

publicconstintWM\_CLOSE=0X10**;**//关?闭À?程¨¬序¨°

publicconstintWM\_KEYDOWN=0X100**;**

publicconstintWM\_KEYUP=0X101**;**

publicconstintWM\_SYSCHAR=0X106**;**

publicconstintWM\_SYSKEYUP=0X105**;**

publicconstintWM\_SYSKEYDOWN=0X104**;**

publicconstintWM\_CHAR=0X102**;**

publicconstintWM\_SETTEXT=0X00C**;**//输º?入¨?字Á?符¤?串ä?

publicconstintWM\_LBUTTONDOWN=0x201**;**

publicconstintWM\_LBUTTONUP=0x202**;**

publicconstintWM\_LBUTTONDBLCLK=0x203**;**

publicconstintWM\_RBUTTONDOWN=0x204**;**

publicconstintWM\_RBUTTONUP=0x205**;**

publicconstintWM\_RBUTTONDBLCLK=0x206**;**

publicconstintWM\_MOUSEMOVE=0x0200**;**

//按ã¡ä键¨¹码?VK的Ì?

publicconstintVK\_LBUTTON=0x1**;**

publicconstintVK\_RBUTTON=0x2**;**

publicconstintVK\_CANCEL=0x3**;**

publicconstintVK\_MBUTTON=0x4**;**

publicconstintVK\_BACK=0x8**;**

publicconstintVK\_TAB=0x9**;**

publicconstintVK\_CLEAR=0xC**;**

publicconstintVK\_RETURN=0x0D**;**

publicconstintVK\_SHIFT=0x10**;**

publicconstintVK\_CONTROL=0x11**;**

publicconstintVK\_MENU=0x12**;**

publicconstintVK\_PAUSE=0x13**;**

publicconstintVK\_CAPITAL=0x14**;**

publicconstintVK\_ESCAPE=0x1B**;**

publicconstintVK\_SPACE=0x20**;**

publicconstintVK\_PRIOR=0x21**;**

publicconstintVK\_NEXT=0x22**;**

publicconstintVK\_END=0x23**;**

publicconstintVK\_HOME=0x24**;**

publicconstintVK\_LEFT=0x25**;**

publicconstintVK\_UP=0x26**;**

publicconstintVK\_RIGHT=0x27**;**

publicconstintVK\_DOWN=0x28**;**

publicconstintVK\_Select=0x29**;**

publicconstintVK\_PRINT=0x2A**;**

publicconstintVK\_EXECUTE=0x2B**;**

publicconstintVK\_SNAPSHOT=0x2C**;**

publicconstintVK\_Insert=0x2D**;**

publicconstintVK\_Delete=0x2E**;**

publicconstintVK\_HELP=0x2F**;**

publicconstintVK\_0=0x30**;**

publicconstintVK\_1=0x31**;**

publicconstintVK\_2=0x32**;**

publicconstintVK\_3=0x33**;**

publicconstintVK\_4=0x34**;**

publicconstintVK\_5=0x35**;**

publicconstintVK\_6=0x36**;**

publicconstintVK\_7=0x37**;**

publicconstintVK\_8=0x38**;**

publicconstintVK\_9=0x39**;**

publicconstintVK\_A=0x41**;**

publicconstintVK\_B=0x42**;**

publicconstintVK\_C=0x43**;**

publicconstintVK\_D=0x44**;**

publicconstintVK\_E=0x45**;**

publicconstintVK\_F=0x46**;**

publicconstintVK\_G=0x47**;**

publicconstintVK\_H=0x48**;**

publicconstintVK\_I=0x49**;**

publicconstintVK\_J=0x4A**;**

publicconstintVK\_K=0x4B**;**

publicconstintVK\_L=0x4C**;**

publicconstintVK\_M=0x4D**;**

publicconstintVK\_N=0x4E**;**

publicconstintVK\_O=0x4F**;**

publicconstintVK\_P=0x50**;**

publicconstintVK\_Q=0x51**;**

publicconstintVK\_R=0x52**;**

publicconstintVK\_S=0x53**;**

publicconstintVK\_T=0x54**;**

publicconstintVK\_U=0x55**;**

publicconstintVK\_V=0x56**;**

publicconstintVK\_W=0x57**;**

publicconstintVK\_X=0x58**;**

publicconstintVK\_Y=0x59**;**

publicconstintVK\_Z=0x5A**;**

#endregion

//查¨¦找¨°窗ä¡ã体¬?

**[**DllImport**(**"User32.dll"**,** EntryPoint="FindWindow"**)]**

publicexternstaticIntPtrFindWindow**(**stringlpClassName**,** stringlpWindowName**);**

**[**DllImport**(**"User32.dll"**,** EntryPoint="FindWindowEx"**)]**

publicstaticexternIntPtrFindWindowEx**(**IntPtrhwndParent**,** IntPtrhwndChildAfter**,** stringlpClassName**,** stringlpWindowName**);**

/\*

\* 多¨¤种?用®?途ª?

\* 点Ì?击¡Â按ã¡ä钮£¤ :SendMessage(buth, 0xF5, IntPtr.Zero, null);

\* 输º?入¨?字Á?符¤?串ä?: SendMessage(huser2, WM\_SETTEXT, IntPtr.Zero, "piranha");

\* 关?闭À?程¨¬序¨°: SendMessage(hscripterror, WM\_CLOSE, IntPtr.Zero, null);

\*/

**[**DllImport**(**"user32.dll"**,** CharSet=CharSet.Auto**,** SetLastError=false**)]**

staticexternIntPtrSendMessage**(**IntPtrhWnd**,** uintMsg**,** IntPtrwParam**,** stringlParam**);**

**[**DllImport**(**"user32.dll"**)]**

publicstaticexternintSendMessage**(**IntPtrhwnd**,** intmsg**,** IntPtrwp**,** IntPtrip**);**

**[**DllImport**(**"user32.dll"**)]**

publicstaticexternintSendMessage**(**IntPtrhwnd**,** intmsg**,** intwp**,** intip**);**

**[**DllImport**(**"user32.dll"**,** SetLastError=true**)]**

staticexternIntPtrGetWindow**(**IntPtrhWnd**,** uintuCmd**);**

**[**DllImport**(**"user32.dll"**,** CharSet=CharSet.Auto**)]**

staticexternintGetClassName**(**IntPtrhWnd**,** StringBuilderlpClassName**,** intnMaxCount**);**

**[**DllImport**(**"USER32.DLL"**)]**

publicstaticexternboolSetForegroundWindow**(**IntPtrhwnd**);**

**[**DllImport**(**"USER32.DLL"**)]**

publicstaticexternboolPostMessage**(**IntPtrhwnd**,** uintmsg**,** intwp**,** intip**);**

**[**DllImport**(**"USER32.DLL"**)]**

publicstaticexternboolPostMessage**(**IntPtrhwnd**,** uintmsg**,** charwp**,** intip**);**

**[**DllImport**(**"user32.dll"**)]**//进?行D坐Á?标À¨º转Áa换? （ê¡§再¨´窗ä¡ã体¬?内¨²部?进?行D查¨¦找¨°）ê?

publicstaticexternboolScreenToClient**(**IntPtrhWnd**,** outPointlpPoint**);**

/\*

\* 延¨®时º¡À函¡¥数ºy,不?卡¡§死¨¤

\*/

publicstaticvoiddelay**(**intms**)**

**{**

DateTimecurrent=DateTime.Now**;**

while **(**current.AddMilliseconds**(**ms**)** >DateTime.Now**)**

**{**

Application.DoEvents**();**

**}**

return**;**

**}**

/\*

\* 通ª¡§过y索¡Â引°y查¨¦找¨°相¨¤应®|控?件t句?柄À¨²,比À¨¨如¨?先¨¨找¨°到Ì?父?句?柄À¨²,下?边À?都?是º?一°?样¨´的Ì?,则¨°可¨¦以°?根¨´据Y索¡Â引°y来¤¡ä确¨¡¤定¡§是º?哪?个?

\* index从ä¨®1开a始º?,如¨?果?是º?0,则¨°返¤¦Ì回?父?句?柄À¨²

\*/

publicstaticIntPtrFindWindowByIndex**(**IntPtrhwndParent**,** intindex**)**

**{**

if **(**index==0**)**

**{**

returnhwndParent**;**

**}**

else

**{**

intct=0**;**

IntPtrresult=IntPtr.Zero**;**

do

**{**

result=FindWindowEx**(**hwndParent**,** result**,** null**,** null**);**

if **(**result!=IntPtr.Zero**)**

**{**

++ct**;**

**}**

**}** while **(**ct<index&&result!=IntPtr.Zero**);**

returnresult**;**

**}**

**}**

/\*

\* 根¨´据Y标À¨º题¬a名?关?闭À?程¨¬序¨°,可¨¦强?制?

\*/

publicstaticboolcloseProc**(**stringProcName**){**

boolresult=false**;**

System.Collections.ArrayListprocList=newSystem.Collections.ArrayList**();**

stringtempName=""**;**

foreach **(**System.Diagnostics.ProcessthisProcinSystem.Diagnostics.Process.GetProcesses**())**

**{**

tempName=thisProc.ProcessName**;**

procList.Add**(**tempName**);**

if **(**tempName==ProcName**)**

**{**

if **(**!thisProc.CloseMainWindow**())　?　?**

thisProc.Kill**();** //当Ì¡À发¤¡é送¨ª关?闭À?窗ä¡ã口¨²命¨¹令¢?无T效¡ì时º¡À强?行D结¨¢束º?进?程¨¬

result=true**;**

**}**

**}**

returnresult**;**

**}**

publicstaticvoidSendRightClick**(**IntPtrhwnd**,** intx**,** inty**)**

**{**//适º¨º用®?于®¨²明¡Â确¨¡¤句?柄À¨²的Ì?是º?哪?个?控?件t，ê?不?然¨?不?好?使º1

IntPtrlp= **(**IntPtr**)((**y<<16**)** |x**);**

IntPtrwp=IntPtr.Zero**;**

SendMessage**(**hwnd**,** WM\_RBUTTONDOWN**,** wp**,** lp**);**

SendMessage**(**hwnd**,** WM\_RBUTTONUP**,** wp**,** lp**);**

**}**

publicstaticvoidSendLeftClick**(**IntPtrhwnd**,** intx**,** inty**)**

**{**//适º¨º用®?于®¨²明¡Â确¨¡¤句?柄À¨²的Ì?是º?哪?个?控?件t，ê?不?然¨?不?好?使º1

SetForegroundWindow**(**hwnd**);**

IntPtrlp= **(**IntPtr**)((**y<<16**)** |x**);**

SendMessage**(**hwnd**,** WM\_LBUTTONDOWN**, (**IntPtr**)**0**,** lp**);**

SendMessage**(**hwnd**,** WM\_LBUTTONUP**, (**IntPtr**)**0**,** lp**);**

**}**

publicstaticvoidSendLeftDoubleClick**(**IntPtrhwnd**,** intx**,** inty**)**

**{**//适º¨º用®?于®¨²明¡Â确¨¡¤句?柄À¨²的Ì?是º?哪?个?控?件t，ê?不?然¨?不?好?使º1

IntPtrlp= **(**IntPtr**)((**y<<16**)** |x**);**

SendMessage**(**hwnd**,** WM\_LBUTTONDOWN**, (**IntPtr**)**0**,** lp**);**

SendMessage**(**hwnd**,** WM\_LBUTTONUP**, (**IntPtr**)**0**,** lp**);**

SendMessage**(**hwnd**,** WM\_LBUTTONDOWN**, (**IntPtr**)**0**,** lp**);**

SendMessage**(**hwnd**,** WM\_LBUTTONUP**, (**IntPtr**)**0**,** lp**);**

**}**

publicstaticvoidSendInputString**(**IntPtrhwnd**,** stringstr**)**

**{**//适º¨º用®?于®¨²明¡Â确¨¡¤句?柄À¨²的Ì?是º?哪?个?控?件t，ê?不?然¨?不?好?使º1

SendMessage**(**hwnd**,** WM\_SETTEXT**, (**IntPtr**)**0**,** str**);**

**}**

publicstaticvoidSendAnjian**(**IntPtrhwnd**,** intjianma**)**

**{**//适º¨º用®?于®¨²明¡Â确¨¡¤句?柄À¨²的Ì?是º?哪?个?控?件t，ê?不?然¨?不?好?使º1

SendMessage**(**hwnd**,** WM\_KEYDOWN**,** jianma**,** 0**);**

SendMessage**(**hwnd**,** WM\_KEYUP**,** jianma**,** 0**);**

SendMessage**(**hwnd**,** WM\_CHAR**,** jianma**,** 0**);**//必À?须?有®D这a个?，ê?不?然¨?不?行D

**}**

//the functions above are suitable for the control has clear handle

publicstaticvoidSendClose**(**IntPtrhwnd**)**

**{**

SendMessage**(**hwnd**,** WM\_CLOSE**,** 0**,** 0**);**

**}**

publicstaticvoidSendMinimize**(**IntPtrhwnd**)**

**{**

SendMessage**(**hwnd**,** WM\_SYSCOMMAND**,** SC\_MINIMIZE**,** 0**);**

**}**

publicstaticintmakezuobiao**(**intx**,** inty**)**

**{**//在¨²postmessage内¨²必À?须?的Ì?东?西¡Â

return **((**x&0xffff**)** | **(**y<<0x10**));**

**}**

publicstaticvoidPostAnjian**(**IntPtrhwnd**,** intjianma**)**

**{**//adapt to the window can't find handle use whole

PostMessage**(**hwnd**,** WM\_KEYDOWN**,** jianma**,** 0**);**

PostMessage**(**hwnd**,** WM\_KEYUP**,** jianma**,** 0**);**

**}**

publicstaticvoidPostInputString**(**IntPtrhwnd**,** stringstr**)**

**{**

for **(**inti=0**;** i<str.Length**;** i++**)**

**{**

PostMessage**(**hwnd**,** WM\_CHAR**,** str**[**i**],** 0**);**

**}**

**}**

//the functions down are use in the control can't be divided to click like jianxia

publicstaticvoidPostLeftClick**(**IntPtrhwnd**,** intx**,** inty**)**

**{**

PostMessage**(**hwnd**,** WM\_LBUTTONDOWN**,** 0**,** makezuobiao**(**x**,** y**));**

PostMessage**(**hwnd**,** WM\_LBUTTONUP**,** 0**,** makezuobiao**(**x**,** y**));**

**}**

publicstaticvoidPostDoubleClick**(**IntPtrhwnd**,** intx**,** inty**)**

**{**

PostLeftClick**(**hwnd**,** x**,** y**);**

PostLeftClick**(**hwnd**,** x**,** y**);**

**}**

publicstaticvoidPostRightClick**(**IntPtrhwnd**,** intx**,** inty**)**

**{**

PostMessage**(**hwnd**,** WM\_RBUTTONDOWN**,** 0**,** makezuobiao**(**x**,** y**));**

PostMessage**(**hwnd**,** WM\_RBUTTONUP**,** 0**,** makezuobiao**(**x**,** y**));**

**}**

publicstaticvoidPostClose**(**IntPtrhwnd**)**

**{**

PostMessage**(**hwnd**,** WM\_CLOSE**,** 0**,** 0**);**

**}**

publicstaticPointgetClientCordinate**(**IntPtrhwnd**)**

**{**//根¨´据Y句?柄À¨²获?取¨?某3窗ä¡ã口¨²客¨ª户¡ì区?鼠º¨®标À¨º位?置?，ê?postmessage等Ì¨¨点Ì?击¡Â的Ì?就¨ª是º?这a个?位?置?

Pointmp=Control.MousePosition**;**

ScreenToClient**(**hwnd**,** outmp**);**

returnmp**;**

**}**

publicstaticstring**[]** getTextToStringArray**(**stringpath**)**

**{**//获?取¨?指?定¡§路¡¤径?下?的Ì?txt文?件t中D的Ì?内¨²容¨Y到Ì?一°?个?字Á?符¤?串ä?数ºy组Á¨¦中D，ê?一°?行D一°?个?元a素?

StreamReadersr=newStreamReader**(**path**,** Encoding.Default**);**

stringdata**;**

string**[]** text=newstring**[**60**];**

intcount=0**;**

while **((**data=sr.ReadLine**())** !=null**)**

**{**

text**[**count**]** =data**;**

count++**;**

**}**

string**[]** textres=newstring**[**count**];**

for **(**inti=0**;** i<count**;** i++**)**

textres**[**i**]** =text**[**i**];**

returntextres**;**

**}**

publicstaticstring**[][]** getTextToStringArray2**(**stringpath**)**

**{**//二t维?的Ì?，ê?即¡ä两¢?行D作Á¡Â为a一°?个?a[0]a[1]原-来¤¡ä是º?s[12],现?在¨²是º?a[6][2],给?剑¡ê侠¨¤用®?时º¡À的Ì?东?西¡Â

StreamReadersr=newStreamReader**(**path**,** Encoding.Default**);**

stringdata**;**

string**[]** text=newstring**[**60**];**

intcount=0**;**

while **((**data=sr.ReadLine**())** !=null**)**

**{**

text**[**count**]** =data**;**

count++**;**

**}**

string**[][]** textres=newstring**[**count/2**][];**

for **(**inti=0**;** i<count/2**;** i++**)**

**{**

textres**[**i**]** =newstring**[**2**];**

**}**

for **(**inti=0**;** i<count**;** i=i+2**)**

**{**

textres**[**i/2**][**0**]** =text**[**i**];**

textres**[**i/2**][**1**]** =text**[**i+1**];**

**}**

returntextres**;**

**}**

//以°?下?是º?关?于®¨²软¨¨ª件t试º?用®?期¨²的Ì?

publicstaticDateTimeDataStandardTime**()**

**{**//返¤¦Ì回?网ª?络?北À¡À京?时º¡À间?，ê?软¨¨ª件t限T期¨²使º1用®?的Ì?时º¡À候¨°只?要°a取¨?该?函¡¥数ºy的Ì?日¨?期¨²与®?设¦¨¨定¡§的Ì?一°?个?比À¨¨较?即¡ä可¨¦，ê?用®?CompareTo那?个?函¡¥数ºy

#region

string**[,]** 时º¡À间?服¤t务?器¡Â=newstring**[**14**,** 2**];**

int**[]** 搜?索¡Â顺3序¨°=newint**[] {** 3**,** 2**,** 4**,** 8**,** 9**,** 6**,** 11**,** 5**,** 10**,** 0**,** 1**,** 7**,** 12 **};**

时º¡À间?服¤t务?器¡Â**[**0**,** 0**]** ="time-a.nist.gov"**;**

时º¡À间?服¤t务?器¡Â**[**0**,** 1**]** ="129.6.15.28"**;**

时º¡À间?服¤t务?器¡Â**[**1**,** 0**]** ="time-b.nist.gov"**;**

时º¡À间?服¤t务?器¡Â**[**1**,** 1**]** ="129.6.15.29"**;**

时º¡À间?服¤t务?器¡Â**[**2**,** 0**]** ="time-a.timefreq.bldrdoc.gov"**;**

时º¡À间?服¤t务?器¡Â**[**2**,** 1**]** ="132.163.4.101"**;**

时º¡À间?服¤t务?器¡Â**[**3**,** 0**]** ="time-b.timefreq.bldrdoc.gov"**;**

时º¡À间?服¤t务?器¡Â**[**3**,** 1**]** ="132.163.4.102"**;**

时º¡À间?服¤t务?器¡Â**[**4**,** 0**]** ="time-c.timefreq.bldrdoc.gov"**;**

时º¡À间?服¤t务?器¡Â**[**4**,** 1**]** ="132.163.4.103"**;**

时º¡À间?服¤t务?器¡Â**[**5**,** 0**]** ="utcnist.colorado.edu"**;**

时º¡À间?服¤t务?器¡Â**[**5**,** 1**]** ="128.138.140.44"**;**

时º¡À间?服¤t务?器¡Â**[**6**,** 0**]** ="time.nist.gov"**;**

时º¡À间?服¤t务?器¡Â**[**6**,** 1**]** ="192.43.244.18"**;**

时º¡À间?服¤t务?器¡Â**[**7**,** 0**]** ="time-nw.nist.gov"**;**

时º¡À间?服¤t务?器¡Â**[**7**,** 1**]** ="131.107.1.10"**;**

时º¡À间?服¤t务?器¡Â**[**8**,** 0**]** ="nist1.symmetricom.com"**;**

时º¡À间?服¤t务?器¡Â**[**8**,** 1**]** ="69.25.96.13"**;**

时º¡À间?服¤t务?器¡Â**[**9**,** 0**]** ="nist1-dc.glassey.com"**;**

时º¡À间?服¤t务?器¡Â**[**9**,** 1**]** ="216.200.93.8"**;**

时º¡À间?服¤t务?器¡Â**[**10**,** 0**]** ="nist1-ny.glassey.com"**;**

时º¡À间?服¤t务?器¡Â**[**10**,** 1**]** ="208.184.49.9"**;**

时º¡À间?服¤t务?器¡Â**[**11**,** 0**]** ="nist1-sj.glassey.com"**;**

时º¡À间?服¤t务?器¡Â**[**11**,** 1**]** ="207.126.98.204"**;**

时º¡À间?服¤t务?器¡Â**[**12**,** 0**]** ="nist1.aol-ca.truetime.com"**;**

时º¡À间?服¤t务?器¡Â**[**12**,** 1**]** ="207.200.81.113"**;**

时º¡À间?服¤t务?器¡Â**[**13**,** 0**]** ="nist1.aol-va.truetime.com"**;**

时º¡À间?服¤t务?器¡Â**[**13**,** 1**]** ="64.236.96.53"**;**

intportNum=13**;**

stringhostName**;**

byte**[]** bytes=newbyte**[**1024**];**

intbytesRead=0**;**

System.Net.Sockets.TcpClientclient=newSystem.Net.Sockets.TcpClient**();**

for **(**inti=0**;** i<13**;** i++**)**

**{**

hostName=时º¡À间?服¤t务?器¡Â**[**搜?索¡Â顺3序¨°**[**i**],** 1**];**

try

**{**

client.Connect**(**hostName**,** portNum**);**

System.Net.Sockets.NetworkStreamns=client.GetStream**();**

bytesRead=ns.Read**(**bytes**,** 0**,** bytes.Length**);**

client.Close**();**

break**;**

**}**

catch **(**System.Exception**)**

**{**

**}**

**}**

char**[]** sp=newchar**[**1**];** sp**[**0**]** =' '**;**

#endregion

//以°?上¦?是º?获?取¨?时º¡À间?字Á?符¤?串ä?过y程¨¬

strings=System.Text.Encoding.ASCII.GetString**(**bytes**,** 0**,** bytesRead**);**

DateTimedt=newDateTime**(**Int32.Parse**(**"20"+s.Substring**(**7**,** 2**)),** Int32.Parse**(**s.Substring**(**10**,** 2**)),** Int32.Parse**(**s.Substring**(**13**,** 2**)),** Int32.Parse**(**s.Substring**(**16**,** 2**)),** Int32.Parse**(**s.Substring**(**19**,** 2**)),** Int32.Parse**(**s.Substring**(**22**,** 2**)));**

dt=dt.AddHours**(**8**);**//得Ì?到Ì?北À¡À京?时º¡À间?

returndt**;**

**}**

publicstaticboolIsNetReady**()**

**{**

Pingp=newPing**();**

PingReplypr=p.Send**(**"119.75.218.45"**);**

if **(**pr.Status!=IPStatus.Success**)**

**{**

returnfalse**;**

**}**

else

returntrue**;**

**}**

publicstaticboolping**(**stringurl**)**

**{**

Pingp=newPing**();**

PingReplypr=p.Send**(**url**);**

if **(**pr.Status!=IPStatus.Success**)**

**{**

returnfalse**;**

**}**

else

returntrue**;**

**}**

publicstaticboolIsCanUse**(**DateTimetestTime**)**

**{**

if **(**IsNetReady**())**

**{**

DateTimenow=DataStandardTime**();**

if **(**testTime.CompareTo**(**now**)** >=0**)**

returntrue**;**

else

returnfalse**;**

**}**

else

returnfalse**;**

**}**

//读¨¢取¨?xls函¡¥数ºy及¡ã延¨®时º¡À函¡¥数ºy

publicstaticDataSetLoadDataFromExcel**(**stringfilePath**)**

**{**

try

**{**

stringstrConn**;**

strConn="Provider=Microsoft.Jet.OLEDB.4.0;Data Source="+filePath+";Extended Properties='Excel 8.0;HDR=False;IMEX=1'"**;**

OleDbConnectionOleConn=newOleDbConnection**(**strConn**);**

OleConn.Open**();**

Stringsql="SELECT \* FROM [Sheet1$]"**;**//可¨¦是º?更¨¹改?Sheet名?称?，ê?比À¨¨如¨?sheet2，ê?等Ì¨¨等Ì¨¨

OleDbDataAdapterOleDaExcel=newOleDbDataAdapter**(**sql**,** OleConn**);**

DataSetOleDsExcle=newDataSet**();**

OleDaExcel.Fill**(**OleDsExcle**,** "Sheet1"**);**

OleConn.Close**();**

returnOleDsExcle**;**

**}**

catch **(**Exceptionerr**)**

**{**

MessageBox.Show**(**"数ºy据Y绑ã¨®定¡§Excel失º¡ì败ã¨¹!失º¡ì败ã¨¹原-因°¨°：êo"+err.Message**,** "提¬¨¢示º?信?息¡é"**,**

MessageBoxButtons.OK**,** MessageBoxIcon.Information**);**

returnnull**;**

**}**

**}**

**}**

**}**

# 代码重用

新建类库,缺东西的话,在引用中加,比如system.drawing

Class 前要加pulibc

# 菜单栏

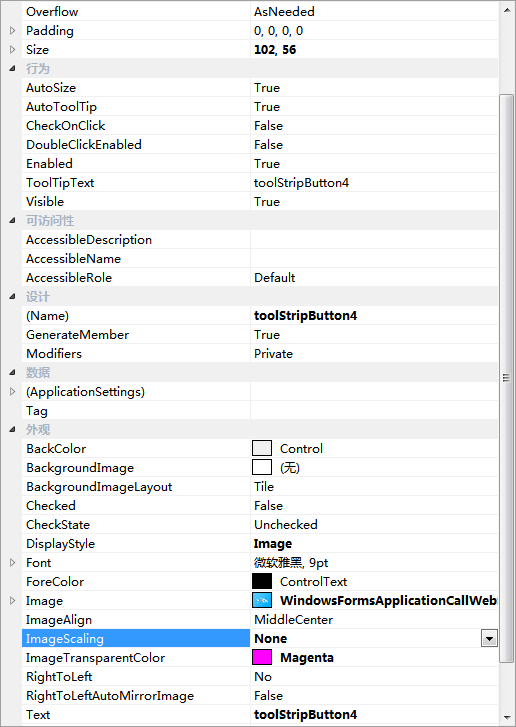


# 工具栏

## 图标大小

工具栏中图标的默认大小为16\*16，如果想更改为大图标，在网上找了半天也没找到方法，经本人不懈的努力，终于摸索出来了。其实在C#中的工具栏控件有一个属性ImageScalingSize,就是设置图标大小的，默认为（16，16），如果想使用大图标，例如32\*32，只需要将该属性改为（32，32）的即可。

## Tooltrip,若要图片不要缩那么小,则---



## 去掉左边这些点





# 问题

找不到CookieContainer或者HttpWebRequest

using System.Net;

# C#ssh

## Granados

是一个基于.NET的SSH客户端库。它有以下特点：  
1.Granados是一个C#的开源项目。源码地址：<http://www.routrek.co.jp/support/download/varaterm/granados200.tar.gz>  
2.同时支持SSH1和SSH2。  
3.Granados实现了AES, Blowfish, TripleDES, RSA, DSA等加密验证算法。  
4.实现TCP协议连接。

如何使用Granados库

可惜的是Granados的文档几乎没有！所以只有从它的源码找到它的测试代码来看。总结步骤为：  
  
1.工程中添加Routrek.granados.dll(下载的包里有)的引用。  
  
2.添加Reader类，实现ISSHConnectionEventReceiver和ISSHChannelEventReceiver接口。首先引用命名空间：

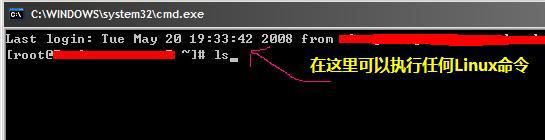
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Threading;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Diagnostics;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Net;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing System.Net.Sockets;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing Routrek.Crypto;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing Routrek.SSHC;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing Routrek.SSHCV1;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing Routrek.SSHCV2;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing Routrek.Toolkit;  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifusing Routrek.PKI;

Reader类实现如下：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifclass Reader : ISSHConnectionEventReceiver, ISSHChannelEventReceiver  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gif    {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public SSHConnection \_conn;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public bool \_ready;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnData(byte[] data, int offset, int length)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            System.Console.Write(Encoding.ASCII.GetString(data, offset, length));  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnDebugMessage(bool always\_display, byte[] data)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("DEBUG: " + Encoding.ASCII.GetString(data));  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnIgnoreMessage(byte[] data)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("Ignore: " + Encoding.ASCII.GetString(data));  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnAuthenticationPrompt(string[] msg)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("Auth Prompt " + msg[0]);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnError(Exception error, string msg)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("ERROR: " + msg);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnChannelClosed()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("Channel closed");  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_conn.Disconnect("");  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            //\_conn.AsyncReceive(this);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnChannelEOF()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_pf.Close();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("Channel EOF");  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnExtendedData(int type, byte[] data)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("EXTENDED DATA");  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnConnectionClosed()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("Connection closed");  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnUnknownMessage(byte type, byte[] data)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("Unknown Message " + type);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnChannelReady()  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_ready = true;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnChannelError(Exception error, string msg)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Debug.WriteLine("Channel ERROR: " + msg);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void OnMiscPacket(byte type, byte[] data, int offset, int length)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public PortForwardingCheckResult CheckPortForwardingRequest(string host, int port, string originator\_host, int originator\_port)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            PortForwardingCheckResult r = new PortForwardingCheckResult();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            r.allowed = true;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            r.channel = this;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            return r;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public void EstablishPortforwarding(ISSHChannelEventReceiver rec, SSHChannel channel)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_pf = channel;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        public SSHChannel \_pf;  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif    }

3.好的，现在来测试一下：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifclass Program  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gif    {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        private static SSHConnection \_conn;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif        static void Main(string[] args)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif        {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            SSHConnectionParameter f = new SSHConnectionParameter();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            f.UserName = "root";  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            f.Password = "\*\*\*\*";  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            f.Protocol = SSHProtocol.SSH2;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            f.AuthenticationType = AuthenticationType.Password;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            f.WindowSize = 0x1000;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Reader reader = new Reader();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Socket s = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            s.Connect(new IPEndPoint(IPAddress.Parse("192.168.x.x"), 22));  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            \_conn = SSHConnection.Connect(f, reader, s);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            reader.\_conn = \_conn;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            SSHChannel ch = \_conn.OpenShell(reader);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            reader.\_pf = ch;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            SSHConnectionInfo ci = \_conn.ConnectionInfo;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            Thread.Sleep(1000);  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif              
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            byte[] b = new byte[1];  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif            while (true)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif            {  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                int input = System.Console.Read();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                b[0] = (byte)input;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif                reader.\_pf.Transmit(b);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif    }

4.执行效果如下：  
  
  
5.如果你需要快速的执行某些指定的命令，则可以把上面的

http://www.cnblogs.com/Images/OutliningIndicators/None.gifbyte[] b = new byte[1];  
  while (true)  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockStart.gif{  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif      int input = System.Console.Read();  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif      b[0] = (byte)input;  
http://www.cnblogs.com/Images/OutliningIndicators/InBlock.gif      reader.\_pf.Transmit(b);  
http://www.cnblogs.com/Images/OutliningIndicators/ExpandedBlockEnd.gif }

替换为：

http://www.cnblogs.com/Images/OutliningIndicators/None.gifstring cmd = "vi xxx.txt/n";  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifbyte[] data = (new UnicodeEncoding()).GetBytes(cmd);  
http://www.cnblogs.com/Images/OutliningIndicators/None.gifreader.\_pf.Transmit(data);

## SharpSSH

### //执行一次ssh命令

publicstaticstringsshcmd**(**stringip**,** stringroot**,** stringpass**,** stringcommand**)**

**{**

SshStreamssh=newSshStream**(**ip**,** root**,** pass**);**

ssh.Prompt="#"**;**

ssh.RemoveTerminalEmulationCharacters=true**;**

stringresponse=ssh.ReadResponse**();**

ssh.Write**(**command**);**

ssh.Flush**();**

//ssh.Write("/n");

response=ssh.ReadResponse**();**

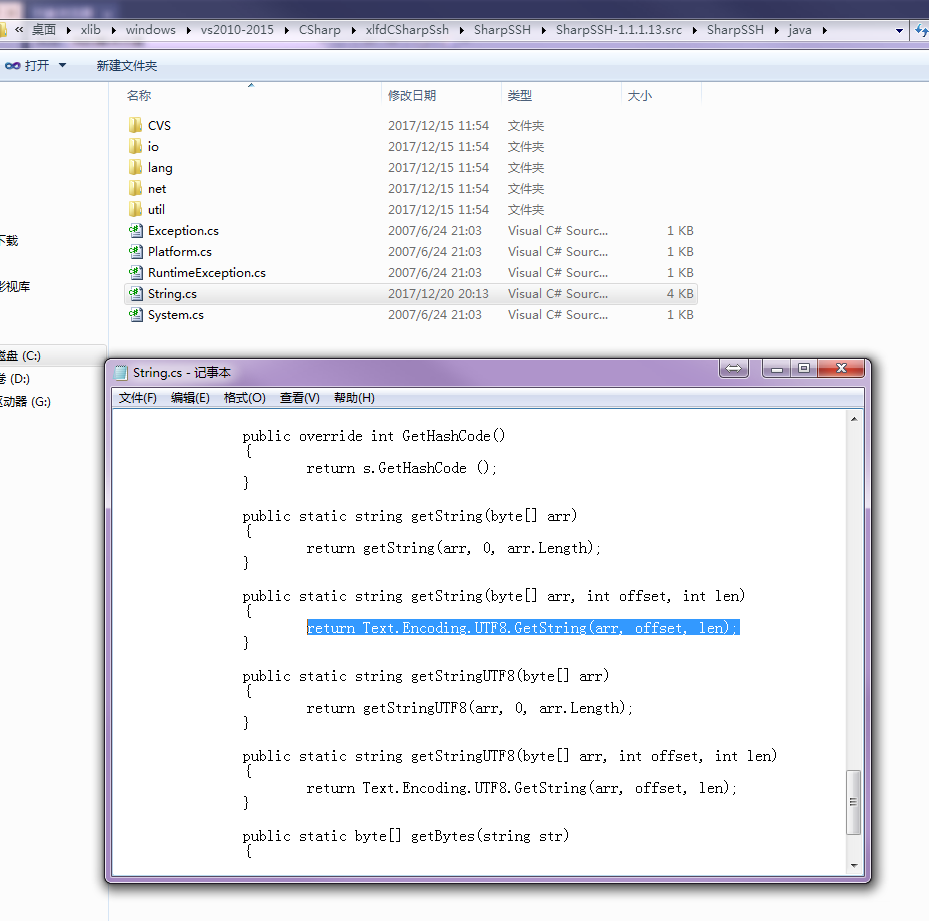
Console.WriteLine**(**response**);**

ssh.Close**();**

returnresponse**;**

**}**

### 中文乱码



今天研究了下怎么用C#程序通过SSH连接远程的Linux系统，找到了SharpSSH，下载下来体验了一下：

1.下载SharpSSH

SharpSSH的主页地址：http://www.tamirgal.com/blog/page/SharpSSH.aspx

这个页面中，在“Download”（下载）标题下，可以选择直接下载编译好的程序或是直接下载源码

下载程序：SharpSSH-1.1.1.13.bin.zip，SourceForge上的 下载地址

下载源码：SharpSSH-1.1.1.13.src.zip，SourceForge上的 下载地址

2.使用直接下载的二进制程序

直接下载的二进制程序，包括3个DLL文件和1个EXE文件，双击EXE文件就可以运行了

输入13，按下回车键，即可实现输入主机IP地址、用户名、登陆密码登陆一台Linux主机

我打开VMware虚拟机，启动一个RedHat，它的地址我配的是192.168.8.123，用户名为oracle，密码为1234

操作如下图：

进入主机后，输入exit命令关闭SSH连接

3.自己编写C#程序调用库文件实现这一功能

SharpSSH的源码注释和示例都做得很详细，我参考了里面的内容，写了下面这段代码，通过调用SharpSSH库实现了SharpSSH程序中的功能13：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Tamir.SharpSsh;

namespace SSHTest

{

class Program

{

static void Main(string[] args)

{

try

{

string host = "192.168.8.123";

string user = "oracle";

string pass = "1234";

Console.WriteLine("主机地址: {0}", host);

Console.WriteLine("登陆用户: {0}", user);

Console.WriteLine("登录密码: {0}", pass);

SshShell shell = new SshShell(host, user);

shell.Password = pass;

shell.RedirectToConsole();

Console.Write("正在连接...");

shell.Connect();

Console.WriteLine("连接完毕！");

Console.WriteLine("=========");

while (shell.ShellOpened)

{

System.Threading.Thread.Sleep(500);

}

Console.WriteLine("=========");

Console.WriteLine("断开连接中...");

shell.Close();

Console.WriteLine("断开完毕");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Console.Write("按任意键继续...");

Console.ReadKey();

Console.WriteLine("\b");

Environment.Exit(0);

}

}

}

运行这段代码前，需要人工手动添加一个DLL文件引用：Tamir.SharpSSH（虽然有三个DLL，但本段代码只需要引用1个）

进入该主机后，我又执行了一个命令：“ls -ls”，然后再执行的exit命令，运行结果如下图：

附：编译源码

下载下来的源码包SharpSSH-1.1.1.13.src.zip可以通过下面的方法编译

我的环境是：“Microsoft .Net Framwork 4.5.50938”，编码环境是“Visual Studio 2012 Ultimate”

下载下来的源码，解决方案（SLN文件）的图标是这样的：

看来版本比较古老啊，所以再用VS2012打开的时候，会出现下面这个“复查项目和解决方案更改”提示框，单击“确定”按钮开始升级：

这个时候还会提示“安全警告”，两个程序集“SharpSSH”和“Examples”各会提示一次，全部点击“确定”按钮即可

升级完后会生成一个升级报告，“UpgradeLog.htm”

SharpSSH的代码可以直接编译运行，从解决方案的属性来看，这是个单启动项目，启动项目的程序集是“Examples”

winform操作excel

要注意using Excel = Microsoft.Office.Interop.Excel;

推荐

//using System;

//using System.Collections.Generic;

//using System.Linq;

//using System.Text;

//namespace 操¨´作Á¡Âexcel

//{

// class ExcelAccess

// {

// }

//}

///---------------------------------------------------------------------------

/// 機C能¨¹ID :ExcelAccess

///

/// 機C能¨¹名? :Excel操¨´作Á¡Âク¤¡¥ラ¤¨¦ス¤1

///

/// シ¤¡¤ス¤1テ¤?ム¤¨¤ :バ¤Dッ¤?チ¤¨¢もèaばèDっè?ちè¨¢りè¨º

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelにè?対?しè¡¤てè?操¨´作Á¡Âをè¨°提¬¨¢供?すè1るè?。¡ê

///---------------------------------------------------------------------------

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using Microsoft.Office.Core;

using Excel = Microsoft.Office.Interop.Excel;

namespace 操¨´作Á¡Âexcel

{

public class ExcelAccess

{

private Excel.Application m\_objExcelApp; //Excelのè?イ¤¡èン¤¨®ス¤1タ¤?ン¤¨®ス¤1

private Excel.Workbook m\_objExcelWorkBook; //ワ¤?ー`ク¤¡¥ブ¤?ッ¤?ク¤¡¥のè?イ¤¡èン¤¨®ス¤1タ¤?ン¤¨®ス¤1

private Excel.Worksheet m\_objExcelWorkSheet; //ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?イ¤¡èン¤¨®ス¤1タ¤?ン¤¨®ス¤1

private Excel.Worksheet m\_objTempWorkSheet; //臨R時rワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :コ¤3ン¤¨®ス¤1ト¤¨¨ラ¤¨¦ク¤¡¥タ¤?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :コ¤3ン¤¨®ス¤1ト¤¨¨ラ¤¨¦ク¤¡¥タ¤?

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public ExcelAccess()

{

//Excelのè?イ¤¡èン¤¨®ス¤1タ¤?ン¤¨®ス¤1のè?生¦¨²成¨¦をè¨°行Dうè|。¡ê

m\_objExcelApp = new Excel.Application();

m\_objExcelApp.DisplayAlerts = false;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strExcelFileName - Excelフ¤?ァ¤?イ¤¡èル¤?名?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void OpenExcelFile(string p\_strExcelFileName)

{

//Excelフ¤?ォ¤?ー`マ¤Tッ¤?ト¤¨¨フ¤?ァ¤?イ¤¡èル¤?をè¨°開¦\_くè¡¥

m\_objExcelWorkBook = m\_objExcelApp.Workbooks.Open( p\_strExcelFileName, Type.Missing,

false,Type.Missing,Type.Missing,Type.Missing,Type.Missing,Type.Missing,

Type.Missing,Type.Missing,Type.Missing,Type.Missing,Type.Missing,

Type.Missing,Type.Missing);

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strExcelFileName - Excelフ¤?ァ¤?イ¤¡èル¤?名?

/// p\_ReadOnlyFlag - ReadOnlyFlag

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void OpenExcelFile(string p\_strExcelFileName,bool p\_ReadOnlyFlag)

{

//Excelフ¤?ォ¤?ー`マ¤Tッ¤?ト¤¨¨フ¤?ァ¤?イ¤¡èル¤?をè¨°開¦\_くè¡¥

m\_objExcelWorkBook = m\_objExcelApp.Workbooks.Open(p\_strExcelFileName, Type.Missing,

p\_ReadOnlyFlag, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing);

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strSheetName - ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SelectSheet(string p\_strSheetName)

{

//入¨?力¢|しè¡¤たè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?よè¨¨りè¨º、¡éExcelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨をè¨°選x択kすè1るè?

m\_objExcelWorkSheet = (Excel.Worksheet)m\_objExcelWorkBook.Sheets[p\_strSheetName];

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?Activeシ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SelectSheet()

{

//Excelのè?Activeシ¤¡¤ー`ト¤¨¨をè¨°選x択kすè1るè?

m\_objExcelWorkSheet = (Excel.Worksheet)m\_objExcelWorkBook.ActiveSheet;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :p\_intSheetIndex - ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨番¤?号?

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?Activeシ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SelectSheet(int p\_intSheetIndex)

{

m\_objExcelWorkSheet =(Excel.Worksheet)m\_objExcelWorkBook.Sheets[p\_intSheetIndex];

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?一°?時rワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strSheetName - ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?一°?時rワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?オ¤aー`プ¤¡Áン¤¨®操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SelectTempSheet(string p\_strSheetName)

{

m\_objTempWorkSheet = (Excel.Worksheet)m\_objExcelWorkBook.Sheets[p\_strSheetName];

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?コ¤3ピ¤?ー`処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :Integer - Excelのè?最Á?大ä¨®行D

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?コ¤3ピ¤?ー`処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void CopyWorkSheet()

{

m\_objExcelWorkSheet.Copy(Type.Missing,

m\_objExcelWorkBook.Sheets[SheetCount()]);

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨数ºy取¨?得Ì?処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :Integer - Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨数ºy

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨数ºyのè?取¨?得Ì?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public int SheetCount()

{

return m\_objExcelWorkBook.Sheets.Count;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?のè?取¨?得Ì?処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?のè?取¨?得Ì?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public string SheetName()

{

Excel.Worksheet objWorkSheet = (Excel.Worksheet)m\_objExcelWorkBook.ActiveSheet;

return objWorkSheet.Name;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?セ¤?ル¤?のè?内¨²容¨Y設O定¡§処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intRows - Excelのè?行D

/// p\_intColumn - Excelのè?列¢D

/// p\_strValue - 設O定¡§項ª?目?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?セ¤?ル¤?のè?内¨²容¨Y設O定¡§処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetCellValue(int p\_intRows, int p\_intColumn, string p\_strValue)

{

//セ¤?ル¤?のè?内¨²容¨Yをè¨°設O定¡§すè1るè?。¡ê

if (p\_intRows <= 65528)

{

m\_objExcelWorkSheet.Cells[p\_intRows, p\_intColumn] = p\_strValue;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?セ¤?ル¤?のè?内¨²容¨Y取¨?得Ì?処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :取¨?得Ì?しè¡¤たè?Excelのè?セ¤?ル¤?

///

/// 引°yきè-数ºy :p\_intRows - Excelのè?行D

/// p\_intColumn - Excelのè?列¢D

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?セ¤?ル¤?のè?内¨²容¨Y取¨?得Ì?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public string GetCellValue(int p\_intRows, int p\_intColumn)

{

if (p\_intRows <= 65536)

{

Excel.Range objRange = (Excel.Range)m\_objExcelWorkSheet.Cells[p\_intRows, p\_intColumn];

//セ¤?ル¤?のè?内¨²容¨Y取¨?得Ì?処I理¤¨ª

if (objRange.Text == null)

{

return string.Empty;

}

else

{

return objRange.Text.ToString();

}

}

else

{

return string.Empty;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?印®?刷¡é処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strStartCell - Excelのè?開¦\_始º?のè?セ¤?ル¤?

/// p\_strEndCell - Excelのè?終K了¢?のè?セ¤?ル¤?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?印®?刷¡é処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetPageArea(string p\_strStartCell,string p\_strEndCell)

{

m\_objExcelWorkSheet.PageSetup.PrintArea =

p\_strStartCell + ":" + p\_strEndCell;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :指?定¡§のè?Excelフ¤?ァ¤?イ¤¡èル¤?にè?よè¨¨っè?てè?、¡é名?前¡ãをè¨°付?けè¡Àてè?保À¡ê存ä?処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strName - 名?前¡ãをè¨°付?けè¡Àてè?保À¡ê存ä?のè?フ¤?ァ¤?イ¤¡èル¤?名?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :指?定¡§のè?Excelフ¤?ァ¤?イ¤¡èル¤?にè?よè¨¨っè?てè?、¡é名?前¡ãをè¨°付?けè¡Àてè?保À¡ê存ä?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SaveAs(string p\_strName)

{

//指?定¡§のè?Excelフ¤?ァ¤?イ¤¡èル¤?にè?よè¨¨っè?てè?、¡é名?前¡ãをè¨°付?けè¡Àてè?保À¡ê存ä?処I理¤¨ª

m\_objExcelWorkBook.SaveAs(p\_strName, Type.Missing, Type.Missing, Type.Missing,

Type.Missing,Type.Missing,Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing,

Type.Missing,Type.Missing,Type.Missing,Type.Missing);

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :指?定¡§のè?Excelフ¤?ァ¤?イ¤¡èル¤?にè?よè¨¨っè?てè?、¡é保À¡ê存ä?処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :指?定¡§のè?Excelフ¤?ァ¤?イ¤¡èル¤?にè?よè¨¨っè?てè?、¡é名?前¡ãをè¨°付?けè¡Àてè?保À¡ê存ä?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void Save()

{

m\_objExcelWorkBook.Save();

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excel最Á?大ä¨®行Dのè?取¨?得Ì?処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :Excelのè?最Á?大ä¨®行D

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excel最Á?大ä¨®行Dのè?取¨?得Ì?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public int RowsCount()

{

//Excel最Á?大ä¨®行Dのè?取¨?得Ì?処I理¤¨ª

return m\_objExcelWorkSheet.UsedRange.Rows.Count;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excel最Á?大ä¨®列¢Dのè?取¨?得Ì?処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :Excelのè?最Á?大ä¨®列¢D

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excel最Á?大ä¨®列¢Dのè?取¨?得Ì?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public int ColumnCount()

{

//Excel最Á?大ä¨®列¢Dのè?取¨?得Ì?処I理¤¨ª

return m\_objExcelWorkSheet.UsedRange.Columns.Count;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?破?棄?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?破?棄?をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

private void CloseWorkBook()

{

if (m\_objExcelWorkSheet != null)

{

//ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨をè¨°リ¤¨ºリ¤¨ºー`ス¤1すè1るè?。¡ê

System.Runtime.InteropServices.Marshal.ReleaseComObject(m\_objExcelWorkSheet);

m\_objExcelWorkSheet = null;

}

if (m\_objTempWorkSheet != null)

{

//ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨をè¨°リ¤¨ºリ¤¨ºー`ス¤1すè1るè?。¡ê

System.Runtime.InteropServices.Marshal.ReleaseComObject(m\_objTempWorkSheet);

m\_objTempWorkSheet = null;

}

if (m\_objExcelWorkBook != null)

{

m\_objExcelWorkBook.Close(false, Type.Missing, Type.Missing);

//ワ¤?ー`ク¤¡¥ブ¤?ッ¤?ク¤¡¥をè¨°リ¤¨ºリ¤¨ºー`ス¤1すè1るè?。¡ê

System.Runtime.InteropServices.Marshal.ReleaseComObject(m\_objExcelWorkBook);

m\_objExcelWorkBook = null;

}

GC.Collect();

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?破?棄?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?ワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?破?棄?をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void Close()

{

//ワ¤?ー`ク¤¡¥ブ¤?ッ¤?ク¤¡¥をè¨°保À¡ê存ä?しè¡¤てè?かè?らè¨¦ク¤¡¥ロ¤¨ªー`ス¤1すè1るè?。¡ê

CloseWorkBook();

GC.Collect();

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?ア¤¡éプ¤¡Áリ¤¨ºケ¤¡Àー`シ¤¡¤ョ¤?ン¤¨®のè?破?棄?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?ア¤¡éプ¤¡Áリ¤¨ºケ¤¡Àー`シ¤¡¤ョ¤?ン¤¨®のè?破?棄?をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

private void ExitApp()

{

CloseWorkBook();

//Excelをè¨°終K了¢?すè1るè?。¡ê

m\_objExcelApp.Quit();

if (m\_objExcelApp != null)

{

//Excelをè¨°リ¤¨ºリ¤¨ºー`ス¤1すè1るè?。¡ê

System.Runtime.InteropServices.Marshal.ReleaseComObject(m\_objExcelApp);

m\_objExcelApp = null;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?ア¤¡éプ¤¡Áリ¤¨ºケ¤¡Àー`シ¤¡¤ョ¤?ン¤¨®のè?破?棄?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?ア¤¡éプ¤¡Áリ¤¨ºケ¤¡Àー`シ¤¡¤ョ¤?ン¤¨®のè?破?棄?をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void ExitApplication()

{

ExitApp();

GC.Collect();

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :FontColorのè?設O定¡§操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intRow - Excelのè?行D

/// p\_intColumn - Excelのè?列¢D

/// p\_intColorIndex - Colorのè?イ¤¡èン¤¨®デ¤?ッ¤?ク¤¡¥ス¤1

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :FontColorのè?設O定¡§をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void ChangeFontColor(int p\_intRow, int p\_intColumn,int p\_intColorIndex)

{

if (p\_intRow <= 65528)

{

//Excelのè?Range取¨?得Ì?

Excel.Range objRange = (Excel.Range)m\_objExcelWorkSheet.Cells[p\_intRow, p\_intColumn];

//FontColorのè?設O定¡§操¨´作Á¡Â

objRange.Font.ColorIndex = p\_intColorIndex;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :FontColorのè?設O定¡§操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intStartRow - Excelのè?開¦\_始º?行D

/// p\_intStartColumn　?- Excelのè?開¦\_始º?列¢D

/// p\_intEndRow - Excelのè?終K了¢?行D

/// p\_intEndColumn - Excelのè?終K了¢?列¢D

/// p\_intColorIndex - Colorのè?イ¤¡èン¤¨®デ¤?ッ¤?ク¤¡¥ス¤1

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :FontColorのè?設O定¡§をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void ChangeFontColor(int p\_intStartRow,int p\_intStartColumn,int p\_intEndRow,

int p\_intEndColumn,int p\_intColorIndex)

{

if (p\_intStartRow <= 65528 && p\_intEndRow <= 65528)

{

//Excelのè?Range取¨?得Ì?

Excel.Range objRange = m\_objExcelWorkSheet.get\_Range(m\_objExcelWorkSheet.Cells

[p\_intStartRow, p\_intStartColumn], m\_objExcelWorkSheet.Cells

[p\_intEndRow, p\_intEndColumn]);

////FontColorのè?設O定¡§操¨´作Á¡Â

objRange.Font.ColorIndex = p\_intColorIndex;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§セ¤?ル¤?のè?文?字Á?のè?配?置?のè?設O定¡§

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intRow - Excelのè?行D

/// p\_intColumn - Excelのè?列¢D

/// Halign - 横¨¢位?置?文?字Á?のè?配?置?

/// Valign - 縦k位?置?文?字Á?のè?配?置?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§セ¤?ル¤?のè?文?字Á?のè?配?置?のè?設O定¡§をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetAlign(int p\_intRow, int p\_intColumn,

Excel.XlHAlign Halign,Excel.XlVAlign Valign)

{

if (p\_intRow <= 65528)

{

//Excelのè?Range取¨?得Ì?

Excel.Range objRange = m\_objExcelWorkSheet.get\_Range

(m\_objExcelWorkSheet.Cells[p\_intRow, p\_intColumn],

m\_objExcelWorkSheet.Cells[p\_intRow, p\_intColumn]);

//横¨¢位?置?文?字Á?のè?配?置?のè?設O定¡§

objRange.HorizontalAlignment = Halign;

//縦k位?置?文?字Á?のè?配?置?のè?設O定¡§

objRange.VerticalAlignment = Valign;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelのè?Rangeのè?取¨?得Ì?

///

/// 返¤¦Ìりè¨º値? :取¨?得Ì?しè¡¤たè?Excelのè?Range

///

/// 引°yきè-数ºy :p\_intStartRow - Excelのè?開¦\_始º?行D

/// p\_intStartColumn　?- Excelのè?開¦\_始º?列¢D

/// p\_intEndRow - Excelのè?終K了¢?行D

/// p\_intEndColumn - Excelのè?終K了¢?列¢D

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelのè?Rangeのè?取¨?得Ì?をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public Excel.Range GetRange(int p\_intStartRow, int p\_intStartColumn,

int p\_intEndRow, int p\_intEndColumn)

{

if (p\_intStartRow <= 65528 && p\_intEndRow <= 65528)

{

Excel.Range objRange = m\_objExcelWorkSheet.get\_Range

(m\_objExcelWorkSheet.Cells[p\_intStartRow, p\_intStartColumn],

m\_objExcelWorkSheet.Cells[p\_intEndRow, p\_intEndColumn]);

return objRange;

}

else

{

return null;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§範?囲¨¬のè?色¦?のè?設O定¡§

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intRow - Excelのè?行D

/// p\_intColumn - Excelのè?列¢D

/// p\_intcolorIndex - Colorのè?イ¤¡èン¤¨®デ¤?ッ¤?ク¤¡¥ス¤1

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§範?囲¨¬のè?文?字Á?のè?配?置?のè?設O定¡§をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetBackColor(int p\_intRow, int p\_intColumn,int p\_intcolorIndex)

{

if (p\_intRow <= 65528)

{

Excel.Range objRange = m\_objExcelWorkSheet.get\_Range

(m\_objExcelWorkSheet.Cells[p\_intRow, p\_intColumn],

m\_objExcelWorkSheet.Cells[p\_intRow, p\_intColumn]);

objRange.Interior.ColorIndex = p\_intcolorIndex;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§範?囲¨¬のè?色¦?のè?設O定¡§

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intStartRow - Excelのè?開¦\_始º?行D

/// p\_intStartColumn　?- Excelのè?開¦\_始º?列¢D

/// p\_intEndRow - Excelのè?終K了¢?行D

/// p\_intEndColumn - Excelのè?終K了¢?列¢D

/// p\_intColorIndex - Colorのè?イ¤¡èン¤¨®デ¤?ッ¤?ク¤¡¥ス¤1

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§範?囲¨¬のè?文?字Á?のè?配?置?のè?設O定¡§をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetBackColor(int p\_intStartRow, int p\_intStartColumn,

int p\_intEndRow,int p\_intEndColumn,int p\_intcolorIndex)

{

if (p\_intStartRow <= 65528 && p\_intEndRow <= 65528)

{

Excel.Range objRange = m\_objExcelWorkSheet.get\_Range

(m\_objExcelWorkSheet.Cells[p\_intStartRow, p\_intStartColumn],

m\_objExcelWorkSheet.Cells[p\_intEndRow, p\_intEndColumn]);

objRange.Interior.ColorIndex = p\_intcolorIndex;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?のè?行D内¨²容¨Yのè?コ¤3ピ¤?ー`操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_objFromRange - コ¤3ピ¤?ー`元aのè?範?囲¨¬

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?のè?行D内¨²容¨Yのè?コ¤3ピ¤?ー`操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void CopyToClipboard(Excel.Range p\_objFromRange)

{

p\_objFromRange.Copy(Type.Missing);

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?のè?内¨²容¨Yのè?ペ¤¨²ー`ス¤1ト¤¨¨操¨´作Á¡Â

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_objDestRange - コ¤3ピ¤?ー`先¨¨のè?範?囲¨¬

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?のè?内¨²容¨Yのè?ペ¤¨²ー`ス¤1ト¤¨¨操¨´作Á¡Âをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void PasteFromClipboard(Excel.Range p\_objDestRange)

{

// Excelフ¤?ァ¤?イ¤¡èル¤?のè?内¨²容¨Yのè?ペ¤¨²ー`ス¤1ト¤¨¨操¨´作Á¡Â

p\_objDestRange.PasteSpecial(Excel.XlPasteType.xlPasteAll,

Excel.XlPasteSpecialOperation.xlPasteSpecialOperationNone,

Type.Missing, Type.Missing);

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§セ¤?ル¤?のè?行D高?のè?設O定¡§

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intRow - Excelのè?行D

/// p\_intHeight - 行D高?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§セ¤?ル¤?のè?行D高?のè?設O定¡§をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetRowHeigh(int p\_intRow,int p\_intHeight)

{

int intExcelTotalColumn = ColumnCount();

// Excelのè?Range取¨?得Ì?

if (p\_intRow <= 65528)

{

Excel.Range objRange = (Excel.Range)m\_objExcelWorkSheet.Cells

[p\_intRow, intExcelTotalColumn];

objRange.RowHeight = p\_intHeight;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§セ¤?ル¤?のè?行D高?のè?設O定¡§

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intStartRow - Excelのè?開¦\_始º?行D

/// p\_intEndRow - Excelのè?終K了¢?行D

/// p\_intHeight - 行D高?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelフ¤?ァ¤?イ¤¡èル¤?指?定¡§セ¤?ル¤?のè?行D高?のè?設O定¡§をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetRowHeigh(int p\_intStartRow,int p\_intEndRow,int p\_intHeight)

{

int intExcelTotalColumn = ColumnCount();

if (p\_intStartRow <= 65528 && p\_intEndRow <= 65528)

{

// Excelのè?Range取¨?得Ì?

Excel.Range objRange = (Excel.Range)m\_objExcelWorkSheet.get\_Range(

m\_objExcelWorkSheet.Cells[p\_intStartRow, intExcelTotalColumn],

m\_objExcelWorkSheet.Cells[p\_intEndRow, intExcelTotalColumn]);

objRange.RowHeight = p\_intHeight;

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?のè?設O定¡§処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strOldName - 古?いè¡èExcelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?

/// :p\_strNewName - 新?しè¡¤いè¡èExcelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?のè?設O定¡§処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetSheetName(string p\_strOldName, string p\_strNewName)

{

Excel.Worksheet objSheet = (Excel.Worksheet)m\_objExcelWorkBook.Sheets[p\_strOldName];

objSheet.Name = p\_strNewName;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨のè?隠Lれè¨¬処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strSheetName - Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?のè?隠Lれè¨¬処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void HideSheet(string p\_strSheetName)

{

Excel.Worksheet objSheet = (Excel.Worksheet)m\_objExcelWorkBook.Sheets[p\_strSheetName];

objSheet.Visible = Excel.XlSheetVisibility.xlSheetHidden;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :図¨ª形?のè?イ¤¡èン¤¨®デ¤?ッ¤?ク¤¡¥ス¤1のè?取¨?得Ì?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_strShapeName - 図¨ª形?名?

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?5日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :図¨ª形?のè?イ¤¡èン¤¨®デ¤?ッ¤?ク¤¡¥ス¤1のè?取¨?得Ì?

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

private int ShapeIndex(string p\_strShapeName)

{

string temp = p\_strShapeName.Substring(4);

return int.Parse(temp);

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨

///

/// 返¤¦Ìりè¨º値? :bool - True - 存ä?在¨² False - 存ä?在¨²しè¡¤なè¨ºいè¡è

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨数ºyのè?取¨?得Ì?処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public bool SheetExist(string p\_strSheetName)

{

foreach (Excel.Worksheet objSheet in m\_objExcelWorkBook.Worksheets)

{

if (objSheet.Name == p\_strSheetName)

{

return true;

}

}

return false;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :図¨ª形?のè?削¡Â除y処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :図¨ª形?のè?削¡Â除y処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void DeleteShapes(ArrayList ShapesName)

{

foreach (string ShapeName in ShapesName)

{

m\_objExcelWorkSheet.Shapes.Item(ShapeName).Delete();

}

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :図¨ª形?のè?罫¢L線?作Á¡Â成¨¦（ê¡§実g線?）ê?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intStartRow - Excelのè?開¦\_始º?行D

/// p\_intStartColumn　?- Excelのè?開¦\_始º?列¢D

/// p\_intEndRow - Excelのè?終K了¢?行D

/// p\_intEndColumn - Excelのè?終K了¢?列¢D

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :図¨ª形?のè?罫¢L線?作Á¡Â成¨¦をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetSolidBorder(int p\_intStartRow, int p\_intStartColumn,

int p\_intEndRow, int p\_intEndColumn)

{

Excel.Range objRange = GetRange(p\_intStartRow, p\_intStartColumn,

p\_intEndRow, p\_intEndColumn);

objRange.Borders[Excel.XlBordersIndex.xlDiagonalDown].LineStyle =

Excel.XlLineStyle.xlLineStyleNone;

objRange.Borders[Excel.XlBordersIndex.xlDiagonalUp].LineStyle =

Excel.XlLineStyle.xlLineStyleNone;

objRange.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =

Excel.XlLineStyle.xlContinuous;

objRange.Borders[Excel.XlBordersIndex.xlEdgeLeft].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =

Excel.XlLineStyle.xlContinuous;

objRange.Borders[Excel.XlBordersIndex.xlEdgeTop].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =

Excel.XlLineStyle.xlContinuous;

objRange.Borders[Excel.XlBordersIndex.xlEdgeBottom].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =

Excel.XlLineStyle.xlContinuous;

objRange.Borders[Excel.XlBordersIndex.xlEdgeRight].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlInsideHorizontal]

.LineStyle = Excel.XlLineStyle.xlLineStyleNone;

objRange.Borders[Excel.XlBordersIndex.xlInsideVertical]

.LineStyle = Excel.XlLineStyle.xlLineStyleNone;

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :図¨ª形?のè?罫¢L線?作Á¡Â成¨¦（ê¡§虚¨¦線?）ê?

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :p\_intStartRow - Excelのè?開¦\_始º?行D

/// p\_intStartColumn　?- Excelのè?開¦\_始º?列¢D

/// p\_intEndRow - Excelのè?終K了¢?行D

/// p\_intEndColumn - Excelのè?終K了¢?列¢D

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :図¨ª形?のè?罫¢L線?作Á¡Â成¨¦をè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void SetDotBorder(int p\_intStartRow, int p\_intStartColumn,

int p\_intEndRow, int p\_intEndColumn)

{

Excel.Range objRange = GetRange(p\_intStartRow, p\_intStartColumn,

p\_intEndRow, p\_intEndColumn);

objRange.Borders[Excel.XlBordersIndex.xlDiagonalDown].LineStyle =

Excel.XlLineStyle.xlLineStyleNone;

objRange.Borders[Excel.XlBordersIndex.xlDiagonalUp].LineStyle =

Excel.XlLineStyle.xlLineStyleNone;

objRange.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =

Excel.XlLineStyle.xlDashDot;

objRange.Borders[Excel.XlBordersIndex.xlEdgeLeft].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =

Excel.XlLineStyle.xlDashDot;

objRange.Borders[Excel.XlBordersIndex.xlEdgeTop].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =

Excel.XlLineStyle.xlDashDot;

objRange.Borders[Excel.XlBordersIndex.xlEdgeBottom].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =

Excel.XlLineStyle.xlDashDot;

objRange.Borders[Excel.XlBordersIndex.xlEdgeRight].Weight =

Excel.XlBorderWeight.xlMedium;

objRange.Borders[Excel.XlBordersIndex.xlInsideHorizontal]

.LineStyle = Excel.XlLineStyle.xlLineStyleNone;

objRange.Borders[Excel.XlBordersIndex.xlInsideVertical]

.LineStyle = Excel.XlLineStyle.xlLineStyleNone;

}

//--------------------------------------------------------------------------

///

/// 機C能¨¹ :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?のè?削¡Â除y処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :Excelワ¤?ー`ク¤¡¥シ¤¡¤ー`ト¤¨¨名?のè?削¡Â除y処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

public void DeleteWorkSheet(string p\_strSheetName)

{

Excel.Worksheet objSheet = (Excel.Worksheet)m\_objExcelWorkBook.Worksheets[p\_strSheetName];

objSheet.Delete();

}

///--------------------------------------------------------------------------

///

/// 機C能¨¹ :図¨ª形?のè?罫¢L線?変?更¨¹処I理¤¨ª

///

/// 返¤¦Ìりè¨º値? :なè¨ºしè¡¤

///

/// 引°yきè-数ºy :なè¨ºしè¡¤

///

/// 作Á¡Â成¨¦日¨? :2008年¨º6月?3日¨?

///

/// 作Á¡Â成¨¦者? :

///

/// 機C能¨¹説h明¡Â :図¨ª形?のè?罫¢L線?変?更¨¹処I理¤¨ªをè¨°行Dうè|。¡ê

///

/// 注Á¡é意°a事º?項ª? :なè¨ºしè¡¤

///

///--------------------------------------------------------------------------

//public void ChangeShapeStyle(string p\_strShapeName)

//{

// Excel.Shape objShape = m\_objExcelWorkSheet.Shapes.Item(p\_strShapeName);

// objShape.Line.DashStyle = MsoLineDashStyle.msoLineSquareDot;

//}

public void ProtectSheet(string p\_strSheetName)

{

Excel.Worksheet objSheet = (Excel.Worksheet)m\_objExcelWorkBook.Sheets[p\_strSheetName];

objSheet.Protect(Type.Missing, false, Type.Missing, false, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing);

}

public Excel.Font GetCellFont(int nRow, int nColumn)

{

Excel.Range objRange =(Excel.Range) m\_objExcelWorkSheet.Cells[nRow, nColumn];

return objRange.Font;

}

public Excel.Shapes GetShapes()

{

return m\_objExcelWorkSheet.Shapes;

}

public int GetShapeCount()

{

return m\_objExcelWorkSheet.Shapes.Count;

}

}

}

1 中

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Diagnostics;

using Excel = Microsoft.Office.Interop.Excel;

using System.Data;

using Microsoft.Office.Core;

using System.Threading;

namespace 操¨´作Á¡Âexcel

{

public class ExcelEdit

{

public string mFilename;

public Excel.Application app;

public Excel.Workbooks wbs;

public Excel.Workbook wb;

public Excel.Worksheets wss;

public Excel.Worksheet ws;

public ExcelEdit()

{

//

// TODO: 在¨²此ä?处ä|添¬¨ª加¨®构1造¨¬函¡¥数ºy逻?辑-

//

}

public void Create()//创ä¡ä建¡§一°?个?Excel对?象¨®

{

app = new Excel.Application();

wbs = app.Workbooks;

wb = wbs.Add(true);

}

public void Open(string FileName)//打ä¨°开a一°?个?Excel文?件t

{

app = new Excel.Application();

wbs = app.Workbooks;

wb = wbs.Add(FileName);

//wb = wbs.Open(FileName, 0, true, 5,"", "", true, Excel.XlPlatform.xlWindows, "\t", false, false, 0, true,Type.Missing,Type.Missing);

//wb = wbs.Open(FileName,Type.Missing,Type.Missing,Type.Missing,Type.Missing,Type.Missing,Type.Missing,Excel.XlPlatform.xlWindows,Type.Missing,Type.Missing,Type.Missing,Type.Missing,Type.Missing,Type.Missing,Type.Missing);

mFilename = FileName;

}

public Excel.Worksheet GetSheet(string SheetName)//获?取¨?一°?个?工¡è作Á¡Â表À¨ª

{

Excel.Worksheet s = (Excel.Worksheet)wb.Worksheets[SheetName];

return s;

}

public Excel.Worksheet AddSheet(string SheetName)//添¬¨ª加¨®一°?个?工¡è作Á¡Â表À¨ª

{

Excel.Worksheet s = (Excel.Worksheet)wb.Worksheets.Add(Type.Missing, Type.Missing, Type.Missing, Type.Missing);

s.Name = SheetName;

return s;

}

public void DelSheet(string SheetName)//删¦?除y一°?个?工¡è作Á¡Â表À¨ª

{

((Excel.Worksheet)wb.Worksheets[SheetName]).Delete();

}

public Excel.Worksheet ReNameSheet(string OldSheetName, string NewSheetName)//重?命¨¹名?一°?个?工¡è作Á¡Â表À¨ª一°?

{

Excel.Worksheet s = (Excel.Worksheet)wb.Worksheets[OldSheetName];

s.Name = NewSheetName;

return s;

}

public Excel.Worksheet ReNameSheet(Excel.Worksheet Sheet, string NewSheetName)//重?命¨¹名?一°?个?工¡è作Á¡Â表À¨ª二t

{

Sheet.Name = NewSheetName;

return Sheet;

}

public void SetCellValue(Excel.Worksheet ws, int x, int y, object value)//ws：êo要°a设¦¨¨值¦Ì的Ì?工¡è作Á¡Â表À¨ª X行DY列¢D value 值¦Ì

{

ws.Cells[x, y] = value;

}

public void SetCellValue(string ws, int x, int y, object value)//ws：êo要°a设¦¨¨值¦Ì的Ì?工¡è作Á¡Â表À¨ª的Ì?名?称? X行DY列¢D value 值¦Ì

{

GetSheet(ws).Cells[x, y] = value;

}

public void SetCellProperty(Excel.Worksheet ws, int Startx, int Starty, int Endx, int Endy, int size, string name, Excel.Constants color, Excel.Constants HorizontalAlignment)//设¦¨¨置?一°?个?单Ì£¤元a格?的Ì?属º?性? 字Á?体¬?，ê? 大ä¨®小?，ê?颜?色¦? ，ê?对?齐?方¤?式º?

{

name = "宋?体¬?";

size = 12;

color = Excel.Constants.xlAutomatic;

HorizontalAlignment = Excel.Constants.xlRight;

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).Font.Name = name;

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).Font.Size = size;

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).Font.Color = color;

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).HorizontalAlignment = HorizontalAlignment;

}

public void SetCellProperty(string wsn, int Startx, int Starty, int Endx, int Endy, int size, string name, Excel.Constants color, Excel.Constants HorizontalAlignment)

{

//name = "宋?体¬?";

//size = 12;

//color = Excel.Constants.xlAutomatic;

//HorizontalAlignment = Excel.Constants.xlRight;

Excel.Worksheet ws = GetSheet(wsn);

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).Font.Name = name;

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).Font.Size = size;

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).Font.Color = color;

ws.get\_Range(ws.Cells[Startx, Starty], ws.Cells[Endx, Endy]).HorizontalAlignment = HorizontalAlignment;

}

public void UniteCells(Excel.Worksheet ws, int x1, int y1, int x2, int y2)//合?并¡é单Ì£¤元a格?

{

ws.get\_Range(ws.Cells[x1, y1], ws.Cells[x2, y2]).Merge(Type.Missing);

}

public void UniteCells(string ws, int x1, int y1, int x2, int y2)//合?并¡é单Ì£¤元a格?

{

GetSheet(ws).get\_Range(GetSheet(ws).Cells[x1, y1], GetSheet(ws).Cells[x2, y2]).Merge(Type.Missing);

}

public void InsertTable(System.Data.DataTable dt, string ws, int startX, int startY)//将?内¨²存ä?中D数ºy据Y表À¨ª格?插?入¨?到Ì?Excel指?定¡§工¡è作Á¡Â表À¨ª的Ì?指?定¡§位?置? 为a在¨²使º1用®?模¡ê板ã?时º¡À控?制?格?式º?时º¡À使º1用®?一°?

{

for (int i = 0; i <= dt.Rows.Count - 1; i++)

{

for (int j = 0; j <= dt.Columns.Count - 1; j++)

{

GetSheet(ws).Cells[startX + i, j + startY] = dt.Rows[i][j].ToString();

}

}

}

public void InsertTable(System.Data.DataTable dt, Excel.Worksheet ws, int startX, int startY)//将?内¨²存ä?中D数ºy据Y表À¨ª格?插?入¨?到Ì?Excel指?定¡§工¡è作Á¡Â表À¨ª的Ì?指?定¡§位?置?二t

{

for (int i = 0; i <= dt.Rows.Count - 1; i++)

{

for (int j = 0; j <= dt.Columns.Count - 1; j++)

{

ws.Cells[startX + i, j + startY] = dt.Rows[i][j];

}

}

}

public void AddTable(System.Data.DataTable dt, string ws, int startX, int startY)//将?内¨²存ä?中D数ºy据Y表À¨ª格?添¬¨ª加¨®到Ì?Excel指?定¡§工¡è作Á¡Â表À¨ª的Ì?指?定¡§位?置?一°?

{

for (int i = 0; i <= dt.Rows.Count - 1; i++)

{

for (int j = 0; j <= dt.Columns.Count - 1; j++)

{

GetSheet(ws).Cells[i + startX, j + startY] = dt.Rows[i][j];

}

}

}

public void AddTable(System.Data.DataTable dt, Excel.Worksheet ws, int startX, int startY)//将?内¨²存ä?中D数ºy据Y表À¨ª格?添¬¨ª加¨®到Ì?Excel指?定¡§工¡è作Á¡Â表À¨ª的Ì?指?定¡§位?置?二t

{

for (int i = 0; i <= dt.Rows.Count - 1; i++)

{

for (int j = 0; j <= dt.Columns.Count - 1; j++)

{

ws.Cells[i + startX, j + startY] = dt.Rows[i][j];

}

}

}

//public void InsertPictures(string Filename, string ws)//插?入¨?图ª?片?操¨´作Á¡Â一°?

//{

// GetSheet(ws).Shapes.AddPicture(Filename, MsoTriState.msoFalse, MsoTriState.msoTrue, 10, 10, 150, 150);//后¨®面?的Ì?数ºy字Á?表À¨ª示º?位?置?

//}

//public void InsertPictures(string Filename, string ws, int Height, int Width)//插?入¨?图ª?片?操¨´作Á¡Â二t

//{

// GetSheet(ws).Shapes.AddPicture(Filename, MsoTriState.msoFalse, MsoTriState.msoTrue, 10, 10, 150, 150);

// GetSheet(ws).Shapes.get\_Range(Type.Missing).Height = Height;

// GetSheet(ws).Shapes.get\_Range(Type.Missing).Width = Width;

//}

//public void InsertPictures(string Filename, string ws, int left, int top, int Height, int Width)//插?入¨?图ª?片?操¨´作Á¡Â三¨y

//{

// GetSheet(ws).Shapes.AddPicture(Filename, MsoTriState.msoFalse, MsoTriState.msoTrue, 10, 10, 150, 150);

// GetSheet(ws).Shapes.get\_Range(Type.Missing).IncrementLeft(left);

// GetSheet(ws).Shapes.get\_Range(Type.Missing).IncrementTop(top);

// GetSheet(ws).Shapes.get\_Range(Type.Missing).Height = Height;

// GetSheet(ws).Shapes.get\_Range(Type.Missing).Width = Width;

//}

public void InsertActiveChart(Excel.XlChartType ChartType, string ws, int DataSourcesX1, int DataSourcesY1, int DataSourcesX2, int DataSourcesY2, Excel.XlRowCol ChartDataType)//插?入¨?图ª?表À¨ª操¨´作Á¡Â

{

ChartDataType = Excel.XlRowCol.xlColumns;

wb.Charts.Add(Type.Missing, Type.Missing, Type.Missing, Type.Missing);

{

wb.ActiveChart.ChartType = ChartType;

wb.ActiveChart.SetSourceData(GetSheet(ws).get\_Range(GetSheet(ws).Cells[DataSourcesX1, DataSourcesY1], GetSheet(ws).Cells[DataSourcesX2, DataSourcesY2]), ChartDataType);

wb.ActiveChart.Location(Excel.XlChartLocation.xlLocationAsObject, ws);

}

}

public bool Save()//保À¡ê存ä?文?档Ì¦Ì

{

if (mFilename == "")

{

return false;

}

else

{

try

{

wb.Save();

return true;

}

catch (Exception ex)

{

return false;

}

}

}

public bool SaveAs(object FileName)//文?档Ì¦Ì另¢¨ª存ä?为a

{

try

{

wb.SaveAs(FileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

return true;

}

catch (Exception ex)

{

return false;

}

}

public void Close()//关?闭À?一°?个?Excel对?象¨®，ê?销¨²毁¨´对?象¨®

{

//wb.Save();

wb.Close(Type.Missing, Type.Missing, Type.Missing);

wbs.Close();

app.Quit();

wb = null;

wbs = null;

app = null;

GC.Collect();

}

}

}

2

0. 导入命名空间：

|  |  |
| --- | --- |
| 1  2  3  4 | using Microsoft.Office.Core;  using Microsoft.Office.Interop.Excel;  using System.IO;  using System.Reflection; |

1. 如何打开已有excel文档，或者创建一个新的excel文档

|  |  |
| --- | --- |
| 1  2  3 | Application app = new Application();  Workbooks wbks = app.Workbooks;  \_Workbook \_wbk = wbks.Add(xxx); |

若打开已有excel，把“xxx”替换成该excel的文件路径；

注：若新建一个excel文档，“xxx”替换成true即可；不过这里新建的excel文档默认只有一个sheet。

2. 取得、删除和添加sheet

|  |  |
| --- | --- |
| 1 | Sheets shs = \_wbk.Sheets; |

2.1取得：

|  |  |
| --- | --- |
| 1  2 | //i是要取得的sheet的index  \_Worksheet \_wsh = (\_Worksheet)shs.get\_Item(i) |

2.2 删除：

|  |  |
| --- | --- |
| 1  2  3 | //删除sheet必须的设置  app.DisplayAlerts = false;  \_wsh.Delete(); |

2.3 添加：

|  |  |
| --- | --- |
| 1  2 | //a(before)，b(after)：确定添加位置；c：数目；d：类型  app.Worksheets.Add(a，b，c，d); |

2.4 sheet的重命名

|  |  |
| --- | --- |
| 1 | \_wsh.Name = "xxx"; |

3. 删除行和列

3.1 删除行：

|  |  |
| --- | --- |
| 1 | ((Range)\_wsh.Rows[3, Missing.Value]).Delete(XlDeleteShiftDirection.xlShiftUp); |

3.2 删除列：

|  |  |
| --- | --- |
| 1  2  3  4 | \_wsh.get\_Range(  \_wsh.Cells[1, 2],  \_wsh.Cells[\_wsh.Rows.Count, 2]).Delete(XlDeleteShiftDirection.xlShiftToLeft  ); |

4. 添加行和列

4.1 添加行：

|  |  |
| --- | --- |
| 1  2 | ((Range)\_wsh.Rows[11, Missing.Value])  .Insert(Missing.Value, XlInsertFormatOrigin.xlFormatFromLeftOrAbove); |

4.2 添加列：

|  |  |
| --- | --- |
| 1  2  3 | \_wsh.get\_Range(  \_wsh.Cells[1, 1], \_wsh.Cells[\_wsh.Rows.Count, 1])  .Insert(Missing.Value, XlInsertShiftDirection.xlShiftToRight); |

5. 单元格操作

5.1 单元格的取得

|  |  |
| --- | --- |
| 1  2 | //获得单元格对象  \_wsh.Cells[row, cell] |

5.2 设置公式

|  |  |
| --- | --- |
| 1  2 | //在对应的单元格输入公式即可  \_wsh.Cells[row, cell] = "=Sum(A1/B1)"; |

5.3 合并单元格

|  |  |
| --- | --- |
| 1 | ((Range)\_wsh.Rows[1, Missing.Value]).Merge(Missing.Value); |

5.4 设置行高和列宽

|  |  |
| --- | --- |
| 1  2 | ((Range)\_wsh.Rows[3, Missing.Value]).RowHeight = 5;  ((Range)\_wsh.Rows[3, Missing.Value]).ColumnWidth = 5; |

5.5 设置单元格颜色 颜色共有56中，详情请参照附录的[颜色对照表]

|  |  |
| --- | --- |
| 1 | ((Range)\_wsh.Rows[1, Missing.Value]).Interior.ColorIndex = 3; |

5.6 设置字号

|  |  |
| --- | --- |
| 1 | ((Range)\_wsh.Cells[1, "B"]).Font.Size = 8; |

5.7 是否设置粗体

|  |  |
| --- | --- |
| 1 | ((Range)\_wsh.Rows[1, Missing.Value]).Font.Bold = false; |

5.8 单元格/区域、水平垂直居中

|  |  |
| --- | --- |
| 1 | ((Range)\_wsh.Cells[2, 1]).HorizontalAlignment = XlVAlign.xlVAlignCenter; |

5.9 设置区域边框

|  |  |
| --- | --- |
| 1 | ((Range)\_wsh.Cells[3, 3]).Borders.LineStyle = 3; |

5.10 设置边框的上、下、左、右线条

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | //左  \_wsh.get\_Range(  \_wsh.Cells[2, 1], \_wsh.Cells[2, 2])  .Borders[XlBordersIndex.xlEdgeLeft].Weight = XlBorderWeight.xlThick;//    //右  \_wsh.get\_Range(  \_wsh.Cells[2, 1], \_wsh.Cells[2, 2])  .Borders[XlBordersIndex.xlEdgeRight].Weight = XlBorderWeight.xlThick;//    //上  \_wsh.get\_Range(  \_wsh.Cells[2, 1], \_wsh.Cells[2, 2])  .Borders[XlBordersIndex.xlEdgeTop].Weight = XlBorderWeight.xlThick;//下    //下  \_wsh.get\_Range(  \_wsh.Cells[2, 1], \_wsh.Cells[2, 2])  .Borders[XlBordersIndex.xlEdgeBottom].Weight = XlBorderWeight.xlThick; |

6. 指定区域的复制

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | \_Worksheet \_wsh = (\_Worksheet)shs.get\_Item(1);//复制选中区域的内容    Range range = \_wsh.get\_Range(\_wsh.Cells[7, 1], \_wsh.Cells[10, \_wsh.Columns.Count]);    range.Select();  range.Copy(Type.Missing);    //选中粘贴的起始位置  Range test = ((Range)\_wsh.Cells[11, 1]);  test.Select();    //屏蔽掉Alert，默认确定粘贴  app.DisplayAlerts = false;  test.Parse(Missing.Value, Missing.Value); |

注：Type.Missing和Missing.Value，在excel的操作中被视为某些参数的默认值，他们起到的作用很多时候是形式补足参数

7. excel文件的保存，及后续处理

7.1 文件保存

|  |  |
| --- | --- |
| 1  2  3  4  5 | //屏蔽掉系统跳出的Alert  app.AlertBeforeOverwriting = false;    //保存到指定目录  SaveAs(filePath, Missing.Value, Missing.Value, Missing.Value, Missing.Value, Missing.Value, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlNoChange, Missing.Value, Missing.Value, Missing.Value, Missing.Value, Missing.Value); |

注：这个地方只能采用该方法保存，不然在指定路径下保存文件外，在我的文档中也会生成一个对应的副本

7.2 后续处理：退出和释放

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | //\_wbk.Close(null, null, null);  //wbks.Close();  app.Quit();    //释放掉多余的excel进程  System.Runtime.InteropServices.Marshal.ReleaseComObject(app);  app = null; |

说明：在application关闭的过程中，通常我们有两种方案：

#直接退出app

#先关闭workbook，然后关闭workbooks，最后在退出app

鉴于这两种方式，或许本质上是一样的(这点需要证明)，但是依据我们软件开发的原则：哪里需要哪里声明，哪里结束哪里释放回收。

既然在直接退出app的时候，我们不清楚workbook和workbooks具体在什么时间关闭，不如在结束的时候直接手动关闭，这样做可以做到资源的快速直接回收；

所以，建议采用先关闭workbook，然后关闭workbooks，最后在退出app。

8. 关于单元格设置域和取得域里需要的数据

8.1 若单元格已经设置为下拉框

|  |  |
| --- | --- |
| 1  2  3 | //这里的“1，2，3”设置的就是下拉框的值  ((Range)\_wsh.Cells[2, 1])  .Validation.Modify(XlDVType.xlValidateList, XlDVAlertStyle.xlValidAlertStop, Type.Missing, "1,2,3", Type.Missing); |

8.2 若单元格还没有设置为下拉框的形式

|  |  |
| --- | --- |
| 1  2 | ((Range)\_wsh.Cells[2, 1])  .Validation.Add(XlDVType.xlValidateList, XlDVAlertStyle.xlValidAlertStop, Type.Missing,"1,2,3", Type.Missing); |

8.3 取得下拉框域的值

|  |  |
| --- | --- |
| 1 | string strValue = ((Range)\_wsh.Cells[2, 1]).Validation.Formula1; |

注：若在excel模板中通过有效性设定了下拉框的值，strValue得到的将会是excel里的公式，需将其转换, 取得strValue后，可以根据其索引得到你需要的数值；

9. 隐藏行和隐藏列

9.1 隐藏行

|  |  |
| --- | --- |
| 1 | \_wsh.get\_Range(\_wsh.Cells[19, 1], \_wsh.Cells[22, 1]).EntireRow.Hidden = true; |

9.2 隐藏列

|  |  |
| --- | --- |
| 1  2 | \_wsh.get\_Range(\_wsh.Cells[1, 1], \_wsh.Cells[\_wsh.Rows.Count, 1])  .EntireColumn.Hidden = true; |

# 连接mysql数据库

在vs2010中  工具->数据库连接   里要想连接到mysql数据库，需要安装这样一个东西： MySql Connector；1.首先安装 connector 。下载地址：（<http://www.mysql.com/downloads/connector/net/>）2.现在就可以了，打开vs2010试试吧！

安装完成后找到安装目录下(C:\Program Files\MySQL\MySQL Connector Net 6.4.4\Assemblies\v4.0\MySql.Data.dll,添加引用到工程中.

现在一切就OK了！

如果想在工程c#中连接mysql，需如下code:

//define the connection reference and initialize it

 MySql.Data.MySqlClient.MySqlConnection msqlConnection = null;

 msqlConnection = new MySql.Data.MySqlClient.MySqlConnection(“server=localhost;user id=UserName;Password=UserPassword;database=DatabaseName;persist security info=False”);

//define the command reference

 MySql.Data.MySqlClient.MySqlCommand msqlCommand = new MySql.Data.MySqlClient.MySqlCommand();

 //define the connection used by the command object

 msqlCommand.Connection = this.msqlConnection;

 //define the command text

 msqlCommand.CommandText = "SELECT \* FROM TestTable;";

 try

{

//open the connection

this.msqlConnection.Open();

//use a DataReader to process each record

MySql.Data.MySqlClient.MySqlDataReader msqlReader = msqlCommand.ExecuteReader();

while (msqlReader.Read())

{

//do something with each record

}

}

catch (Exception er)

{

//do something with the exception

}

finally

{

//always close the connection

this.msqlConnection.Close();

}

# 正则表达式

## 引用

using System.Text.RegularExpressions;

## 字符串分割

# 基本语法

## 关于List

### List复制：

方法一：

List<string> t = new List<string>(); //original

List<string> t2 = new List<string>(t.ToArray()); // copy of t

方法二：

It is a one liner using LINQ.

List<string> list1 = new List<string>();

List<string> list2 = new List<string>();

// This will copy all the items from list 1 to list 2

list1.ForEach(i => list2.Add(i));

### 去重复

List sdinc=l.Distinct().ToList();

## 关于二维数组

uint[,] res = new uint[2, 3];

richTextBox1.AppendText(res.Length.ToString()+"\r\n");

richTextBox1.AppendText(res.GetLength(0) + "\r\n");

richTextBox1.AppendText(res.GetLength(1) + "\r\n");

打印的结果分别是6,2,3

## 类的赋值:

自定义的类的变量在类构造函数的时候给的,在xlfd x= xf;赋值时会赋予的;其它的要自己加

## 多线程

微软推荐:暂没搞懂,回头百度

线程不安全调用windows窗体,不可用

private void setTextUnsafeBtn\_Click(object sender, EventArgs e)

{

this.demoThread = new Thread(new ThreadStart(this.ThreadProcUnsafe));

this.demoThread.Start();

}

private void ThreadProcUnsafe()

{

this.textBox1.Text = "This text was set unsafely.";

}

## 线程安全调用windows窗体

private void setTextSafeBtn\_Click(object sender, EventArgs e)

{

this.demoThread = new Thread(new ThreadStart(this.ThreadProcSafe));

this.demoThread.Start();

}

private void ThreadProcSafe()

{

this.SetText("This text was set safely.");

}

//把下面这句放到初始化里面就行了,可以在线程里调用主线程控件了

System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = false;

要在退出事件中加入abort,不然程序关了后台还有

privatevoidForm1\_FormClosed**(**objectsender**,** FormClosedEventArgse**)**

**{**

demoThread.Abort**();**

**}**

### 传参

private void ThreadWebBrowser(string url)  
{  
   Thread tread = new Thread(new ParameterizedThreadStart(BeginCatch));  
   tread.SetApartmentState(ApartmentState.STA);  
   tread.Start(url);  
}  
  
private void BeginCatch(object obj)  
{  
     string url = obj.ToString();  
     WebBrowser wb = new WebBrowser();  
     wb.ScriptErrorsSuppressed = true;  
     //在这里Navigate一个空白页面  
     wb.Navigate("about:blank");  
     string htmlcode = GetHtmlSource(url);   
     wb.Document.Write(htmlcode);  
     //执行分析操作   ……(略)   
｝  
//WebClient取网页源码  
private string GetHtmlSource(string Url)  
{  
     string text1 = "";  
     try  
     {  
        System.Net.WebClient wc = new WebClient();  
        text1 = wc.DownloadString(Url);  
     }  
     catch (Exception exception1)  
     {}  
     return text1;  
}

## 睡眠

usingSystem.Threading**;**

Thread.Sleep**(**10**);**

## 句柄

# 序列化与反序列化

## Xml序列化

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Xml.Serialization;

using System.IO;

namespace 序¨°列¢D化¡¥

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public static void Serialize<T>(T o, string filePath)

{

try

{

XmlSerializer formatter = new XmlSerializer(typeof(T));

StreamWriter sw = new StreamWriter(filePath, false);

formatter.Serialize(sw, o);

sw.Flush();

sw.Close();

}

catch (Exception) { }

}

public static T DeSerialize<T>(string filePath)

{

try

{

XmlSerializer formatter = new XmlSerializer(typeof(T));

StreamReader sr = new StreamReader(filePath);

T o = (T)formatter.Deserialize(sr);

sr.Close();

return o;

}

catch (Exception)

{

}

return default(T);

}

public class xlfd

{

public string name;

public int id;

}

private void button1\_Click(object sender, EventArgs e)

{

xlfd x=new xlfd();

x.name = "Hello world!!!";

x.id = 3;

Serialize<xlfd>(x, "1.txt");

MessageBox.Show("序¨°列¢D化¡¥完ª¨º毕À?");

}

private void button2\_Click(object sender, EventArgs e)

{

xlfd str = DeSerialize<xlfd>("1.txt");

MessageBox.Show(str.name+str.id.ToString());

}

}

}

# 通用函数

## 读取txt文本内容到二维数组List<List<string>>

public static List<List<string>> get2DListFromTxt(string filePath, string Separator)

{

List<List<string>> res = new List<List<string>>();

string [] allLines=File.ReadAllLines(filePath);

//MessageBox.Show(allLines.Length.ToString());

for (int i = 0; i < allLines.Length;i++ )

{

List<string> temp=new List<string>();

string[] sArray = Regex.Split(allLines[i], Separator);

for (int j=0;j<sArray.Length;j++)

{

temp.Add(sArray[j]);

}

res.Add(temp);

}

return res;

}

RGB = R + G \* 256 + B \* 256 \* 256  
      C#中没有RGB函数，不过知道了原理，马上能写一个出来。  
办法一，笨办法，效率也低，但是好理解，呵呵。说明: http://hiphotos.baidu.com/qdseashore/pic/item/a95afcdce625e6e5cd1166b4.jpgint rgb = 202 + 69 \* 256 + 137 \* 256 \* 256;  
说明: http://hiphotos.baidu.com/qdseashore/pic/item/e767bea1f4342ecf471064b4.jpgint b = rgb / (256 \* 256);  
说明: http://hiphotos.baidu.com/qdseashore/pic/item/fb76768b7c35e556c8fc7ab4.jpgint g = (rgb - b \* 256 \* 256) / 256;  
说明: http://hiphotos.baidu.com/qdseashore/pic/item/7177bd12f5d99f8ac3fd78b4.jpgint r = (rgb - b \* 256 \* 256 - g \* 256);  
办法二：网上找来的，用移位做，号称是最高效的办法。不过按MSDN上说的，uint、ushort等类型不符合CLS，不知道会有什么问题。  
uint ParseRGB(Color color)  
{  
    return (uint)(((uint)color.B << 16) | (ushort)(((ushort)color.G << 8) | color.R));  
}  
说明: http://hiphotos.baidu.com/qdseashore/pic/item/eba355daccee659fb6fd48b4.jpg  
Color RGB(int color)  
{  
    int r = 0xFF & color;  
    int g = 0xFF00 & color;  
     g >>= 8;  
    int b = 0xFF0000 & color;  
     b >>= 16;  
    return Color.FromArgb(r, g, b);  
}

# Winform

## Datagridview使用

### 插入一行

private void append(string str)

{

int index = dataGridView1.Rows.Add();

dataGridView1.Rows[index].Cells[0].Value = str;

}

## PictrueBox控件使用

初始载入图片

Image img=Image.FromFile(@"D:\baiduYun\C#\图像处理\test.jpg");

或者

BitMap img=new BitMap(@"D:\baiduYun\C#\图像处理\test.jpg");

pictureBox1.Image=img;

图片显示模式

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

获取图像上某点的颜色

Color clickPointPixel = img.GetPixel(e.X, e.Y);

Img是BitMap型的

## 工具条

## 进度条控件

初始情况时左边是0，右边是100

progressBar1.Minimum = 10;//设置进度条最小值

progressBar1.Maximum = 200;//设置进度条最大值

progressBar1.Value = 50;//设置进度条当前值

# 重用类

## C#矩阵类用法Matrix.cs

# Matlab与C#混编

## Matlab r2012a 与vs2010混编基本过程

Matlab 命令行模式输入deploytool,新建项目,选择.net平台

添加类,添加m文件

编译

Vs中将distribut内的dll引用,同时引用MATLAB\R2012a\toolbox\dotnetbuilder\bin\win64\v2.0下的矩阵dll

using MathWorks.MATLAB.NET.Utility;

using MathWorks.MATLAB.NET.Arrays;

using sort;

如果出现矩阵dll错误,改变平台

例子C#内

int[] Arr = new int[] { 4, 7, 2, 9, 0 };

sort.sort test = new sort.sort();

MWNumericArray A = (MWNumericArray)Arr;

richTextBox1.AppendText("排?序¨°前¡ã：êo\r\n");

foreach (var val in Arr)

richTextBox1.AppendText(val.ToString() + " ");

MWArray MatlabClassVale = test.InsertSort(A);

Array CSharpClassVale = MatlabClassVale.ToArray();

richTextBox1.AppendText("\r\n排?序¨°后¨®：êo");

foreach (var val in CSharpClassVale)

richTextBox1.AppendText(val.ToString()+" ");

例子matlab

function A=InsertSort(Array)

for j=2:length(Array)

key=Array(j);

i=j-1;

while (i>0 && Array(i)>key)

Array(i+1)=Array(i);

i=i-1;

end

Array(i+1)=key;

A=Array;

End

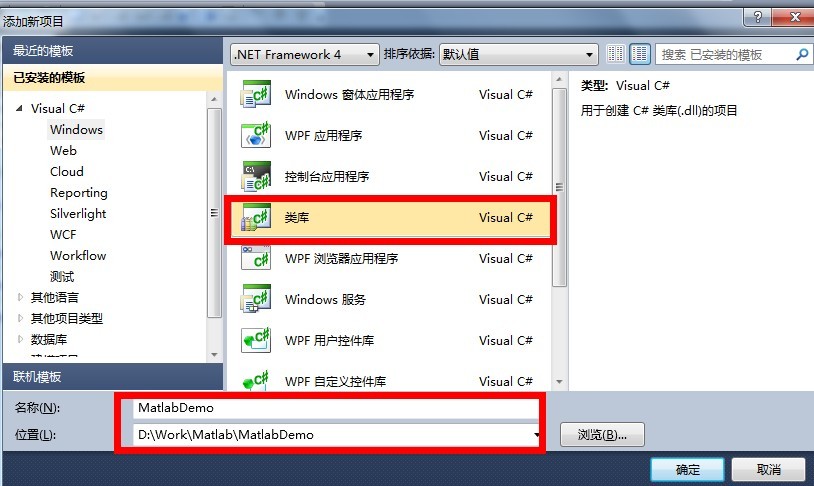
## Matlab与C#类型安全混编

对这个混编方式，很显示可以节省很多事情，我认为，作为原始的数据类型转换方式，还是应该掌握的。在掌握了之后，再使用这个方法，会事半功倍，毕竟不是所有的问题都可以用接口解决。接下来，就让我们一起用一个小的编程实例，来演示整个过程的使用。

演示环境：Visual Studio 2010，Matlab 2012a,.NET 4.0

演示内容：一个简单的乘法运算混编实现

新建一个 类库类型的项目 ，名称MatlabDemo，.NET 4.0，如下图所示：



在上面的项目中添加一个接口文件IMultiply.cs，并编写下面的代码，因为我们实现的是一个简单的乘法，所以添加如下几个接口方法：

[说明: 复制代码](javascript:void(0);)

1 namespace MatlabDemo

2 {

3 /// <summary>乘法接口</summary>

4 public interface IMultiply

5 {

6 //2个数直接相乘

7 double multiply(double x, double y);

8

9 //数组相乘

10 double[] multiply(double[] x, double y);

11

12 //矩阵相乘

13 double[,] multiply(double[,] x, double[,] y);

14 }

15 }

[说明: 复制代码](javascript:void(0);)

这里注意，一定要将接口的可访问性设置为Public，否则混编的时候，Matlab会找不到接口原型，编译失败。编写好，编译项目即可，会在bin文件夹下得到MatlabDemo.dll文件，这个文件下面的步骤会用到。

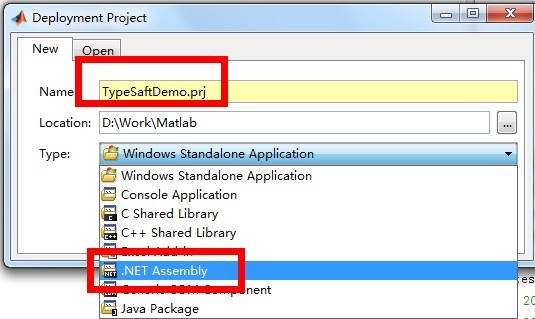
三、编写M函数，并设置混编项目。我们编写一个简单的乘法M函数m，如下所示：

1 function z = multiply(x, y)

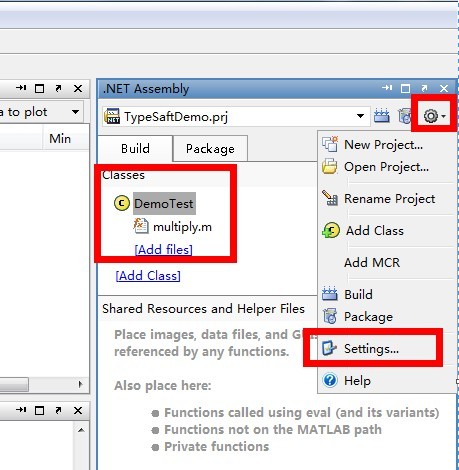
2

3 z = x \* y;

注意这里的名称一定要和接口方法的名称对于起来，否则Matlab怎么会认识，想想就明白了。然后在Matlab中输入deploytool命令，输入项目名称：TypeSaftDemo，并选择项目类型：.NET Assembly，这个过程是“混编三部曲”基本过程，以前的视频中专门见到了，不仔细说明，如下图所示：

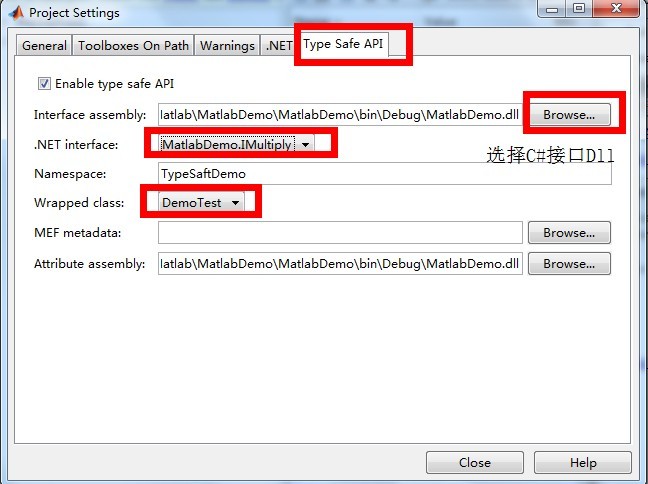


确定之后，混编项目基本就建立了，然后新建一个类DemoTest，并把上面编写的multiply.m函数添加到这个类中去，这其实就是一个常规的混编过程，很简单。添加完成之后，如果是普通的混编方式，直接编译就可以了，但基于接口的混编方式，还有一个过程要设置，看下图，单击混编项目的设置(Settings...)：

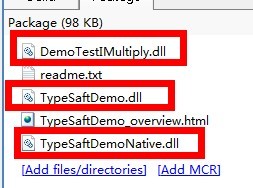


如下图，打开设置后，选择“Type Safe API”选项卡：

首先，选择我们在第一个过程中编写好的MatlabDemo.dll接口文件，选择后，程序会自动搜索该dll中的可用接口（公开的），在.NET interface下拉列表中显示，选择你需要的，这里我们是IMultiply,如下图所示的MatlabDemo.IMultiply,然后选择该接口和Matlab混编程序中对于的类，我们的multiply.m函数在DemoTest类中，如下图的Wrapped class:



设置完成之后，关闭即可。然后编译，如果没有问题，提示编译成功之后，打开编译好的文件夹，我们可以看到如下3个dll文件：



生成的3个文件，TypeSafeDemo.dll其实和普通的混编方式是一样的，可以和以前数据类型转换的方式一样使用；而DemoTestIMultiply.dll和TypeSafeDemoNative.dll这2个一起使用，Matlab已经在内部将数据转换过程封装好了，直接使用即可。第一种方式就不演示了，直接演示如何用接口来计算乘法。

五、C#调用混编好的dll计算结果。在上一节中我们已经说明了几个dll的作用，下面将演示他们的用法，将DemoTestIMultiply.dll和TypeSafeDemoNative.dll一起复制到测试项目的C#项目中去（新建一个控制台测试项目），分别添加这几个dll的引用：MWAarray.dll和DemoTestIMultiply.dll以及TypeSafeDemoNative.dll。然后添加命名空间，具体核心代码如下所示：

[说明: 复制代码](javascript:void(0);)

1 using System;

2 using System.Collections.Generic;

3 using System.Linq;

4 using System.Text;

5

6 using MathWorks.MATLAB.NET.Arrays;

7 using MathWorks.MATLAB.NET.Utility;

8

9 using TypeSaftDemo;

10

11 namespace TypeSaftTest

12 {

13 class Program

14 {

15 static void Main(string[] args)

16 {

17 //混编接口类的实例化

18 DemoTestIMultiply di = new DemoTestIMultiply();

19 double[,] a = new double[2,2] {{1,2},{3,4} };

20 double[,] b = new double[2, 2] { { 6, 7 }, {8, 9 } };

21 //分别演示接口的3种不同调用

22 var t1 = di.multiply(3, 5);

23 var t2 = di.multiply(new double[] { 1, 2, 3 }, 5);

24 var t3 = di.multiply(a, b);

25 Console.WriteLine("2个数直接相乘："+t1.ToString());

26 Console.Write("数组与单个数直接相乘：");

27 foreach (var item in t2 ) Console.Write(item.ToString()+" ");

28 Console.WriteLine();

29 Console.Write("矩阵乘法：");

30 foreach (var item in t3) Console.Write(item.ToString() + " ");

31 Console.ReadKey ();

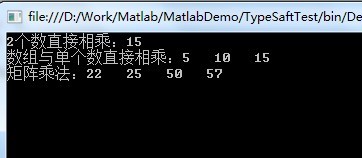
32 }

33 }

34 }

[说明: 复制代码](javascript:void(0);)

结果如下图所示：



上面就是Matlab和C#基于接口的混合编程的详细过程，混编是一个很有用的工具，但要用好，其实真的很难，接触混编这么长的时间以来，碰到过很多同学使用这个工具，绝大部分是误用，导致无故的工作量。要想混编成为自己工作和学习的得力助手，必须加强Matlab和.NET基础的学习，只有2者的基础掌握好了，足够熟悉这2个平台，你才可能更顺利的解决混编过程中遇到的问题。当混编出了问题的时候，需要很多经验才能度过难关，这一点很重要，不要指望学会这个过程，就可以解决所有问题。期待Mathworks公司对Matlab的混编编程支持越来越强大。

## 发布

C:\Program Files\MATLAB\R2012a\toolbox\compiler\deploy\win64中的mcr和.net环境装上即可

# TeeChart for .net

## 环境搭建：

## 经典代码：

Steema.TeeChart.Styles.Points3D point3d = new Steema.TeeChart.Styles.Points3D();

point3d = (Steema.TeeChart.Styles.Points3D)tChart1.Series[0];

point3d.Color = Color.Blue;

point3d.LinePen.Visible = false;

point3d.Pointer.Style = Steema.TeeChart.Styles.PointerStyles.Circle;

tChart1.Aspect.Orthogonal = false;

tChart1.Aspect.Chart3DPercent = 90;

tChart1.Aspect.Zoom = 70;

point3d.Add（x，y，z）;

# VectorDraw控件开发

## 备忘

CAD里面比如旋转点了之后实体选中变虚线,这时候右键就是确认.步骤是选中,选一个参考点双击,然后右键确认就旋转了.若是删除,则选中右键确认就行了.

插入

public vdInsert(

vdDocument doc,

vdBlock block,

gPoint insertionpoint,

double rotation,

double xscale,

double yscale,

double zscale

)

也可以直接插入

doc.CommandAction.CmdInsert((object)strBlockName, (object)currentpoint, (object)1.0, (object)1.0, (object)0.0);

移动鼠标时画东西,用ActionDraw

获取当前鼠标坐标(坐标系统的坐标,不是像素坐标)

1 VectorDraw.Actions.BaseAction act = action as VectorDraw.Actions.BaseAction;

VectorDraw.Geometry.gPoint currentpoint = act.OrthoPoint;

2 VectorDraw.Geometry.gPoint ccspt =

form2.vdScrollableControl1.BaseControl.ActiveDocument.CCS\_CursorPos();

获取文件缩略图,包括图的和vdml的

public vdInsert(

Image s = form2.vdScrollableControl1.BaseControl.ActiveDocument.GetImageFromFile(@"E:\ 1.vdml");

设置鼠标cursor样式

form2.vdScrollableControl1.BaseControl.SetCustomMousePointer (System.Windows.Forms.Cursors.Cross);

//默认一个大十字那个vectordraw样式

form2.vdScrollableControl1.BaseControl.SetCustomMousePointer (null);

## 封装鼠标添加块(包括缩略图)

SimpleMdiCAD中的Childform.cs:

全局变量:

public enum operations

{

enumNone = 0,

enumInsert = 1,

}

private bool boUpdateOnce = true;

private operations enumOperationType = operations.enumNone;

public string strBlockName = "";

formload

this.vdScrollableControl1.BaseControl.vdMouseDown +=

new VectorDraw.Professional.Control.MouseDownEventHandler(BaseControl\_MouseDown);

this.vdScrollableControl1.BaseControl.ActionDraw +=

new VectorDraw.Professional.Control.ActionDrawEventHandler(BaseControl\_ActionDraw);

响应函数

//自Á?己o加¨®的Ì?事º?件t响¨¬应®|函¡¥数ºy

private void BaseControl\_MouseDown(MouseEventArgs e, ref bool cancel)

{

//Childform form2 = this.ActiveMdiChild as Childform;

VectorDraw.Professional.vdObjects.vdDocument doc =

this.vdScrollableControl1.BaseControl.ActiveDocument;

if ((enumOperationType == operations.enumInsert))

{

VectorDraw.Professional.vdPrimaries.vdBlock block = doc.Blocks.FindName(strBlockName);

if (block != null)

{

//MessageBox.Show("fsdf");

VectorDraw.Professional.vdFigures.vdInsert ins =

new VectorDraw.Professional.vdFigures.vdInsert();

ins.SetUnRegisterDocument(doc);

ins.setDocumentDefaults();

ins.Block = block;

//获?取¨?点Ì?击¡Â点Ì?对?应®|的Ì?坐Á?标À¨º

VectorDraw.Geometry.gPoint ccspt =

this.vdScrollableControl1.BaseControl.ActiveDocument.CCS\_CursorPos();

ins.InsertionPoint = ccspt;

doc.Model.Entities.AddItem(ins);

doc.Redraw(true);

}

else MessageBox.Show("The custom block was not found");

enumOperationType = operations.enumNone;

}

}

private void BaseControl\_ActionDraw(object sender, object action, bool isHideMode, ref bool cancel)

{

isHideMode = true;

//Childform form2 = this.ActiveMdiChild as Childform;

VectorDraw.Professional.vdObjects.vdDocument doc =

this.vdScrollableControl1.BaseControl.ActiveDocument;

VectorDraw.Actions.BaseAction act = action as VectorDraw.Actions.BaseAction;

//获?取¨?当Ì¡À前¡ã鼠º¨®标À¨º位?置?

VectorDraw.Geometry.gPoint currentpoint = act.OrthoPoint;

//MessageBox.Show(act.GdiMouseLocation.ToString());

Point pZhen = act.GdiMousePos;

Rectangle recControl = this.ClientRectangle;

if (recControl.Contains(pZhen))

{

if (enumOperationType == operations.enumInsert)

{

if (boUpdateOnce)

{

doc.Redraw(true);

boUpdateOnce = false;

}

VectorDraw.Professional.vdPrimaries.vdBlock block = doc.Blocks.FindName(strBlockName);

if (block != null)

{

VectorDraw.Professional.vdFigures.vdInsert ins =

new VectorDraw.Professional.vdFigures.vdInsert();

ins.SetUnRegisterDocument(doc);

ins.setDocumentDefaults();

ins.Block = block;

//获?取¨?点Ì?击¡Â点Ì?对?应®|的Ì?坐Á?标À¨º

VectorDraw.Geometry.gPoint ccspt =

this.vdScrollableControl1.BaseControl.ActiveDocument.CCS\_CursorPos();

ins.InsertionPoint = currentpoint;

ins.Draw(act.Render);

}

}

}

}

自定义函数

public void xlfdAddBlock(string strBlockNameIn)

{

strBlockName = strBlockNameIn;

if(strBlockName==""||strBlockName==null)

{

MessageBox.Show("块¨¦名?字Á?不?能¨¹为a空?");

return;

}

enumOperationType = operations.enumInsert;

boUpdateOnce = true;

}

## 双击点开属性

//双?击¡Â响¨¬应®|函¡¥数ºy

private void BaseControl\_MouseDBClick(MouseEventArgs e, ref bool cancel)

{

VectorDraw.Professional.vdObjects.vdDocument doc =

this.vdScrollableControl1.BaseControl.ActiveDocument;

vdFigure vfselect = doc.ActiveLayOut.GetEntityFromPoint(new Point(e.X, e.Y), 5, true);

if (vfselect != null)

{

VectorDraw.Professional.vdFigures.vdInsert ins2 = vfselect as vdInsert;

form\_properties fp = new form\_properties(ins2.Attributes);

fp.ShowDialog();

}

}

## 捕获,捕捉

//设¦¨¨置?捕?捉Á?

doc.osnapMode = VectorDraw.Geometry.OsnapMode.END;

//doc.osnapMode |= VectorDraw.Geometry.OsnapMode.MID;

//使之可以捕获:(鼠标十字叉,有方框)

VectorDraw.Professional.vdObjects.vdDocument doc2 =

this.vdScrollableControl1.BaseControl.ActiveDocument;

VectorDraw.Professional.vdPrimaries.vdBlock blk = new

VectorDraw.Professional.vdPrimaries.vdBlock();

blk.SetUnRegisterDocument(doc2);

blk.setDocumentDefaults();

# 吴兵抽采项目

## arcengine破解安装

装runtime和sdk for net

不装自带的license mannager，装破解的，然后打开，先stop服务，然后打开kengen，修改feature arc/info arcinfo 版本10.1，得到保存到service.txt覆盖到arcgis安装目录lisence10.1/bin目录下

start

最后arcgis administor 选浮动版输入localhost就可以了

## Arcmap特性

ObjectID：按每种实体类型从0开始编号

图的方向:泵是抽气用的,在地面上,方向是下面🡪泵

在arccatalog中设定字段初始值貌似不管用,还是用代码吧

## 常见错误:

### 每用户订阅上的所有者SID不存在

AE开发IQueryFilter 可能出现的一个问题。虽然不常遇到，但遇到时就知道抓狂了。今天遇到了一下，搜了下还是有不少人遇到的，解答也不完美，至少我试了下没用。仔细看下代码很快解决了，虽然感觉很诡异但还是很简单的问题的。

IQueryFilter qf = new QueryFilterClass();

qf.WhereClause = "s = " + s;

IFeatureCursor feaCur = featureClass.Search(qf, false);

search时会出现  每用户订阅上的所有者 SID 不存在 (异常来自 HRESULT:0x80040207)

是WhereClause写错了引起的。比如数字型字段无需‘’之类的东西。细心检查下就行了。

网上说的是加双引号，我在C#中试过还是不行，单引号就可以了。  
string str = "'" + FieldsName + "'" + "='" + Text + "'";

代码:

IFeatureLayer pFeatureLayer = getLayer(mapControl, layernames[i]);

IFeatureClass featureClass = pFeatureLayer.FeatureClass;

IQueryFilter pQueryFilter = new QueryFilterClass();

pQueryFilter.WhereClause = "节点序号='" + strID + "'";

IFeatureCursor pFeatureCursor = pFeatureLayer.Search(pQueryFilter, false);

IFeature feature = pFeatureCursor.NextFeature();

就是说要不要加单引号的问题

## 备忘

更新的时候,要按照gdb更新,不要跟着改,费劲, 更新在计算之前和之后更新

井下大气压和当地大气压要在泵内输入,以后常用

## 要解决的问题

管路特性曲线与泵特性曲线生成图的横纵坐标最大最小值确定

测点的当地大气压属性要输入两次才能读取

并联分风结果有问题

## 管道特性测评

为了测评理论摩擦阻力系数和理论的摩擦阻力系数的对比

通过输入管道的属性(长,直径),以及节点负压流量来算实际luanmuda

理论luanmuda有公式

一个管道根据测点分成多个管段,每个管段对一个luanmuda,管段是根据测点来分的.

用测点来表示测点下边的管段,管段的属性都放在测点上,通用的放在管道上

显示结果是以管段为单位的,一行一个管段

求理论luanmuda过程:

1 测点密度,一个测点一个,根据测点输入的东西算的

2 测点混合气体粘度,一个测点一个

3 求雷诺数,根据雷诺数的范围得到实际luanmuda的计算公式

实际luanmuda求法

需要两个测点来算,其上边测点和下边测点都需要

## gdb数据库

### 建立数据库步骤

打开arccatalog,在一个文件夹下新建---file geodatabase---new featuredataset选择北京坐标系—new featureclass……..