# Winfrom各控件起名

//数据类型 数据类型简写 标准命名举例

Label lbl lblMessage

LinkLabel llbl llblToday

Button btn btnSave

TextBox txt txtName

MainMenu mmnu mmnuFile

CheckBox chk chkStock

RadioButton rbtn rbtnSelected

GroupBox gbx gbxMain

PictureBox pic picImage

Panel pnl pnlBody

DataGrid dgrd dgrdView

ListBox lst lstProducts

CheckedListBox clst clstChecked

ComboBox cbo cboMenu

ListView lvw lvwBrowser

TreeView tvw tvwType

TabControl tctl tctlSelected

DateTimePicker dtp dtpStartDate

HscrollBar hsb hsbImage

VscrollBar vsb vsbImage

Timer tmr tmrCount

ImageList ilst ilstImage

ToolBar tlb tlbManage

StatusBar stb stbFootPrint

OpenFileDialog odlg odlgFile

SaveFileDialog sdlg sdlgSave

FoldBrowserDialog fbdlg fgdlgBrowser

FontDialog fdlg fdlgFoot

ColorDialog cdlg cdlgColor

PrintDialog pdlg pdlgPrint

# 用户回车响应事件

第一种：

private void txt\_confirmPwd\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

btn\_submit\_Click(btn\_submit, null);

}

}

第二种：

private void txt\_confirmPwd\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

btn\_submit\_Click(btn\_submit, null);

}

}

# 获取当前系统时间

DateTime dt = DateTime.Now;

dt.ToString();//2005-11-5 13:21:25

dt.ToFileTime().ToString();//127756416859912816

dt.ToFileTimeUtc().ToString();//127756704859912816

dt.ToLocalTime().ToString();//2005-11-5 21:21:25

dt.ToLongDateString().ToString();//2005年¨º11月?5日¨?

dt.ToLongTimeString().ToString();//13:21:25

dt.ToOADate().ToString();//38661.5565508218

dt.ToShortDateString().ToString();//2005-11-5

dt.ToShortTimeString().ToString();//13:21

dt.ToUniversalTime().ToString();//2005-11-5 5:21:25

dt.Year.ToString();//2005

dt.Date.ToString();//2005-11-5 0:00:00

dt.DayOfWeek.ToString();//Saturday

dt.DayOfYear.ToString();//309

dt.Hour.ToString();//13

dt.Millisecond.ToString();//441

dt.Minute.ToString();//30

dt.Month.ToString();//11

dt.Second.ToString();//28

dt.Ticks.ToString();//632667942284412864

dt.TimeOfDay.ToString();//13:30:28.4412864

dt.ToString();//2005-11-5 13:47:04

dt.AddYears(1).ToString();//2006-11-5 13:47:04

dt.AddDays(1.1).ToString();//2005-11-6 16:11:04

dt.AddHours(1.1).ToString();//2005-11-5 14:53:04

dt.AddMilliseconds(1.1).ToString();//2005-11-5 13:47:04

dt.AddMonths(1).ToString();//2005-12-5 13:47:04

dt.AddSeconds(1.1).ToString();//2005-11-5 13:47:05

dt.AddMinutes(1.1).ToString();//2005-11-5 13:48:10

dt.AddTicks(1000).ToString();//2005-11-5 13:47:04

dt.CompareTo(dt).ToString();//0

dt.Add(?).ToString();//问¨º号?为a一°?个?时º¡À间?段?

dt.Equals("2005-11-6 16:11:04").ToString();//False

dt.Equals(dt).ToString();//True

dt.GetHashCode().ToString();//1474088234

dt.GetType().ToString();//System.DateTime

dt.GetTypeCode().ToString();//DateTime

dt.GetDateTimeFormats('s')[0].ToString();//2005-11-05T14:06:25

dt.GetDateTimeFormats('t')[0].ToString();//14:06

dt.GetDateTimeFormats('y')[0].ToString();//2005年¨º11月?

dt.GetDateTimeFormats('D')[0].ToString();//2005年¨º11月?5日¨?

dt.GetDateTimeFormats('D')[1].ToString();//2005 11 05

dt.GetDateTimeFormats('D')[2].ToString();//星?期¨²六¢¨´ 2005 11 05

dt.GetDateTimeFormats('D')[3].ToString();//星?期¨²六¢¨´ 2005年¨º11月?5日¨?

dt.GetDateTimeFormats('M')[0].ToString();//11月?5日¨?

dt.GetDateTimeFormats('f')[0].ToString();//2005年¨º11月?5日¨? 14:06

dt.GetDateTimeFormats('g')[0].ToString();//2005-11-5 14:06

dt.GetDateTimeFormats('r')[0].ToString();//Sat, 05 Nov 2005 14:06:25 GMT

string.Format("{0:d}",dt);//2005-11-5

string.Format("{0:D}",dt);//2005年¨º11月?5日¨?

string.Format("{0:f}",dt);//2005年¨º11月?5日¨? 14:23

string.Format("{0:F}",dt);//2005年¨º11月?5日¨? 14:23:23

string.Format("{0:g}",dt);//2005-11-5 14:23

string.Format("{0:G}",dt);//2005-11-5 14:23:23

string.Format("{0:M}",dt);//11月?5日¨?

string.Format("{0:R}",dt);//Sat, 05 Nov 2005 14:23:23 GMT

string.Format("{0:s}",dt);//2005-11-05T14:23:23

string.Format("{0:t}",dt);//14:23

string.Format("{0:T}",dt);//14:23:23

string.Format("{0:u}",dt);//2005-11-05 14:23:23Z

string.Format("{0:U}",dt);//2005年¨º11月?5日¨? 6:23:23

string.Format("{0:Y}",dt);//2005年¨º11月?

string.Format("{0}",dt);//2005-11-5 14:23:23

string.Format("{0:yyyyMMddHHmmssffff}",dt);

DateTime.Now.ToString(“¡ãyyyy年¨ºMM月?dd日¨?”¡À);//xxxx年¨ºxx月?xx日¨?

# 调用dll文件

长城总结：

1. 如果是.net平台的dll文件，可以直接引用，然后添加引用空间后 就可以直接使用了。
2. 如果是非托管类型的dll文件，需要将dll文件放到windows/system32里面或者程序的Debug/Release里面，然后可以在Programs里面添加如：

using System.Runtime.InteropServices;

public class TSCLIB\_DLL

{

[DllImport("TSCLIB.dll", EntryPoint = "openport")]

public static extern int openport(string printername);

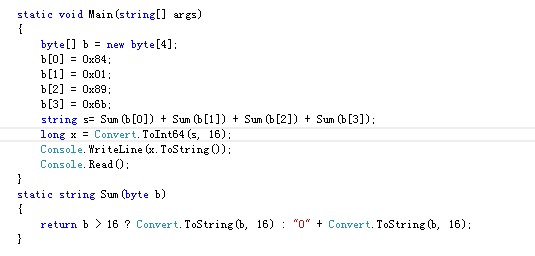
}

然后在调用处 编写如下代码：

TSCLIB\_DLL.openport("TSC TTP-344M Plus");

# 计算器中十六进制转换十进制

第一中方法



第二种方法：

Int32 a = (Int32)cmdlineACK[(Int32)Index.buffer + 3];

Int32 b = (Int32)(cmdlineACK[(Int32)Index.buffer + 2] << 8);

Int32 c = (Int32)(cmdlineACK[(Int32)Index.buffer + 1] << 16);

Int32 d = (Int32)(cmdlineACK[(Int32)Index.buffer] << 24);

ChipID = (Int32)(a + b + c + d);

txt\_ChipID.Text = Convert.ToInt64(Convert.ToString(ChipID, 16).ToString(), 16).ToString();

# 判断是否有选中的项

## 判断一组RadioButton

private string returninfo()

{

string s = "";

foreach (Control c in groupBox2.Controls)

{

if (c is RadioButton)

{ if ((c as RadioButton).Checked) { s += c.Text; } }

}

return s;

}

## 判断菜单

foreach (ToolStripMenuItem men in contextMenuStrip1.Items)

{

foreach (ToolStripMenuItem a in men.DropDownItems)

{

if (a.Checked == true)

{ a.Checked = false; }

}

}

# 进制转换

/// 将字符串转换为相应的进制串

/// <param name="srcInput">输入需要转换的进制字符串</param>

/// <param name="fromBase">源字符串进制</param>

/// <param name="toBase">转换后的进制</param>

public static string HexConvert(string srcInput, int fromBase, int toBase)

{

try

{

long srcChange = Convert.ToInt64(srcInput, fromBase);

return Convert.ToString(srcChange, toBase);

}

catch (Exception)

{ return "进制转换有误."; }

}

//苑斌写的大门口十进制和十六进制互转：

private void ToOX()//转换成十六进制

{

string str= Convert.ToString(Convert.ToInt64(this.cardID10), 16);

if (str.Length > 6)

{

str = str.Substring(str.Length - 6);

}

this.CardID16 = str.PadLeft(8, '0');

}

private void ToOD()//转换成十进制

{

this.cardID10 =Convert.ToInt32(this.cardID16,16).ToString().PadLeft(8,'0');

}

# 退出系统的正规写法

在主窗体添加FormClosing事件

private void gf\_main\_FormClosing(object sender, FormClosingEventArgs e)

{

e.Cancel = (HDIC\_Message.ShowQuestionDialog(this, "是否确定退出系统?") != DialogResult.OK);

}

# 将一个字符串转化为窗体类

//获取窗体的Text值

Type fType = Type.GetType(txt\_subMenu.Text);

Form f = (Form)fType.InvokeMember("随便写", System.Reflection.BindingFlags.CreateInstance, null, null, null);

string formName = f.Text;

# 利用xml设置app.config文件

例如：修改<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<connectionStrings>

<add name ="connStringName" connectionString ="Data Source=.;Initial Catalog=STBInfo;Persist Security Info=False;User ID=sa; pwd =sa"

providerName="System.Data.SqlClient" />

</connectionStrings>

</configuration>

代码可以这样写：

public static void SetValue(string AppKey, string AppValue)

{

XmlDocument xDoc = new XmlDocument();

xDoc.Load(Application.StartupPath+@"\soc\_nds\_csharp.exe.config");

XmlNode xNode;

XmlElement xElem1;

XmlElement xElem2;

xNode = xDoc.SelectSingleNode("//connectionStrings");

xElem1 = (XmlElement)xNode.SelectSingleNode("//add[@name='" + AppKey + "']");

if (xElem1 != null)

{

xElem1.SetAttribute("connectionString", AppValue);

}

else

{

xElem2 = xDoc.CreateElement("add");

xElem2.SetAttribute("name", AppKey);

xElem2.SetAttribute("connectionString", AppValue);

xNode.AppendChild(xElem2);

}

xDoc.Save(Application.StartupPath+@"\soc\_nds\_csharp.exe.config");

}

【读取代码如下】：

xNode = xDoc.SelectSingleNode("//appSettings");

xElem = (XmlElement)xNode.SelectSingleNode("//add[@key=''" + appKey + "'']");

if (xElem != null)

return xElem.GetAttribute("value");

else

return "";

# 将xml字符串转换为datatable

/// <param name="xmlData">xml字符串</param>

/// <param name="tagName">节点名称形如：//ROWSET/ROW</param>

/// <returns>datatable</returns>

public DataTable XMLToDataTable(string xmlData, string tagName)

{

XmlDocument Xmldoc = new XmlDocument();

Xmldoc.LoadXml(xmlData);

XmlNodeList xlist = Xmldoc.SelectNodes(tagName);//參數錯誤

DataTable dt = new DataTable();

DataRow dr;

for (int i = 0; i < xlist.Count; i++)

{

dr = dt.NewRow();

XmlElement xe = (XmlElement)xlist.Item(i);

for (int j = 0; j < xe.Attributes.Count; j++)

{

if (!dt.Columns.Contains("@" + xe.Attributes[j].Name))

{

dt.Columns.Add("@" + xe.Attributes[j].Name);

}

dr["@" + xe.Attributes[j].Name] = xe.Attributes[j].Value;

}

for (int j = 0; j < xe.ChildNodes.Count; j++)

{

if (!dt.Columns.Contains(xe.ChildNodes.Item(j).Name))

{

dt.Columns.Add(xe.ChildNodes.Item(j).Name);

}

dr[xe.ChildNodes.Item(j).Name] = xe.ChildNodes.Item(j).InnerText;

}

dt.Rows.Add(dr);

}

return dt;

}

# [uint--byte[]--char[]--string相互转换汇总](http://www.cnblogs.com/rudy/archive/2009/05/21/1486672.html)

在在做一些互操作的时候往往需要一些类型的相互转换，比如用c#访问win32api的时候往往需要向api中传入DWORD参数 即：uint参数这些数值所表示的数据在实际的应用中可能需要以字符的形式显示，但是c#对api的关系无法跟c++相比，所以在c#中进行一些类型数据的转换十分必要了，  
    下面将用到的一些简单的转换操作贴上来，方便记忆

uint--->byte[]

       byte[] bpara =System.BitConverter.GetBytes(uint upara);

byte[]--->uint

       uint upara= System.BitConverter.ToUint32(bpara);

byte--->char

       system.convert.tochar(bpara);

char--->byte

       system.convert.tobyte(cpara);

byte[]--->char[]

      (1)char[] cpara= System.Text.Encoding.Default.GetChars(bpara);(1)

(2)char[] cpara=new char[bpara.length];

for(int i=0;i <bpara.length;i ++){char[i]=system.convert.tochar(bpara[i]);}

      (3)char[] cpara= new ASCIIEncoding().GetChars(bpara);

char[]--->byte[]

      (1)byte[] bpara= System.Text.Encoding.Default.GetBytes(cpara);

      (2) byte[] bpara=   new ASCIIEncoding().GetBytes(cpara);

char[]--->string

      String spara=new String(cpara);

string---char[]

     char[] cpara=spara.ToCharArray()

uint---->char[]

      (1)uint-->byte[];

      (2)byte[]-->char[];

uint--->string

     (1)uint-->byte[];

     (2)byte[]-->char[];

     (3)char[]-->string;

byte[]--->string

    (1).byte[]-->char[];

    (2).char[]-->string;

   (3) new ASCIIEncoding().GetString(bprar);

char[]--->uint

   (1).char[]-->byte[];

   (2).byte[]-->uint;

string--->byte[]

    bpara= System.Text.Encoding.Default.GetBytes(sPara)

string--->uint

     (1)string-->byte[];

     (2)byte[]-->uint;

    注意在跟api用uint进行字符交互的时候，一定要注意字符顺序，涉及到api中高低位数据的问题，即获取到api中DOWRD的数据在c#表示中往往是反序，所以在c#中获取或者传递字符串时一定要注意反序处理后才能转换成uint给api使用，有机会好好总结一下贴上来。

# 读写xml文件

【首先是长城总结的经验：如果SelectSingleNode("//baoshi ")是一级目录则“//”可以不写，否则不写只能获取的是空值】

------------------------------------自动创建xml文件（可以根据自己要求创建不同节点）---------------------------------------------------------

// --------------------------xml文件开头写上<?xml version="1.0" encoding="utf-8" ?> 或者<?xml version="1.0" ?>

XmlDocument doc = new XmlDocument();

doc.LoadXml("<?xml version='1.0' ?>" +

"<update>" +

"<system>" +

"<baoshi>" + "不报时" + "</baoshi>" +

"</system>" +

"</update>");

// Save the document to a file.

doc.Save(Application.StartupPath + @"\time.xml");

//DataSet ds = new DataSet();//这是第二种自动创建xml的方法

//ds.WriteXml(Application.StartupPath + "/timelist111.xml");

---------------------------------------------------------------【第一种方法】-------------------------------------------

------------已知有一个XML文件（bookstore.xml）如下------------------

<?xml version="1.0" encoding="gb2312"?>

<bookstore>

<book genre="fantasy" ISBN="2-3631-4">

<title>Oberon's Legacy</title>

<author>Corets, Eva</author>

<price>5.95</price>

</book>

</bookstore>

--------------------添加------------------------------------------

XmlDocument xmlDoc = new XmlDocument();

xmlDoc.Load("bookstore.xml");

//xmlDoc.LoadXml("<!--l version=\"1.0\" encoding=\"gb2312\--><BOOKSTORE></BOOKSTORE>");

XmlNode root = xmlDoc.SelectSingleNode("bookstore");//查找<BOOKSTORE></BOOKSTORE>

XmlElement xe1 = xmlDoc.CreateElement("book");//创建一个<BOOK></BOOK>节点

xe1.SetAttribute("genre", "李赞红");//设置该节点genre属性

xe1.SetAttribute("ISBN", "2-3631-4");//设置该节点ISBN属性

XmlElement xesub1 = xmlDoc.CreateElement("title");

xesub1.InnerText = "CS从入门到精通";//设置文本节点

xe1.AppendChild(xesub1);//添加到<BOOK></BOOK>节点中

XmlElement xesub2 = xmlDoc.CreateElement("author");

xesub2.InnerText = "候捷";

xe1.AppendChild(xesub2);

XmlElement xesub3 = xmlDoc.CreateElement("price");

xesub3.InnerText = "58.3";

xe1.AppendChild(xesub3);

root.AppendChild(xe1);//添加到<BOOKSTORE></BOOKSTORE>节点中

xmlDoc.Save("bookstore.xml");

--------------------------修改-----------------------------------

修改节点：将genre属性值为“李赞红“的节点的genre值改为“update李赞红”，将该节点的子节点<author>的文本修改为“亚胜；

XmlNodeList nodeList = xmlDoc.SelectSingleNode("bookstore").ChildNodes;//获取bookstore节点的所有子节点

foreach (XmlNode xn in nodeList)//遍历所有子节点

{

XmlElement xe = (XmlElement)xn;//将子节点类型转换为XmlElement类型

if (xe.GetAttribute("genre") == "李赞红")//如果genre属性值为“李赞红”

{

xe.SetAttribute("genre", "update李赞红");//则修改该属性为“update李赞红”

XmlNodeList nls = xe.ChildNodes;//继续获取xe子节点的所有子节点

foreach (XmlNode xn1 in nls)//遍历

{

XmlElement xe2 = (XmlElement)xn1;//转换类型

if (xe2.Name == "author")//如果找到

{

xe2.InnerText = "亚胜";//则修改

break;//找到退出来就可以了

}

}

break;

}

}

xmlDoc.Save("bookstore.xml");//保存。

----------------------------------删除-----------------------------

删除 <book genre="fantasy" ISBN="2-3631-4">节点的genre属性，删除 <book genre="update李赞红" ISBN="2-3631-4">节点；

XmlNodeList xnl = xmlDoc.SelectSingleNode("bookstore").ChildNodes;

foreach (XmlNode xn in xnl)

{

XmlElement xe = (XmlElement)xn;

if (xe.GetAttribute("genre") == "fantasy")

{

xe.RemoveAttribute("genre");//删除genre属性

}

else if (xe.GetAttribute("genre") == "update李赞红")

{

xe.RemoveAll();//删除该节点的全部内容

}

}

xmlDoc.Save("bookstore.xml");

-----------------------------【第二个版本】---------------------------------------------------------

//参考网站 【http://www.cnblogs.com/wayfarer/archive/2004/07/16/24783.html】

原型：<?xml version="1.0"?>

<UserLogin>

<User>

<UserCode>001</UserCode>

<UserName>操作员1</UserName>

<UserPwd>111</UserPwd>

</User>

<User>

<UserCode>002</UserCode>

<UserName>操作员2</UserName>

<UserPwd>222</UserPwd>

</User>

</UserLogin>

-----------------------------------添加--------------------------------------------------

接着新建项目--类，取名为XmlRW.cs，存放到app\_Code文件夹中

在最上部加上对xml的using : using System.Xml 代码如下：

-------------------添加----------------------------------------------------------

//WriteXml 完成对User的添加操作

//FileName 当前xml文件的存放位置

//UserCode 欲添加用户的编码

//UserName 欲添加用户的姓名

//UserPassword 欲添加用户的密码

public void WriteXML(string FileName,string UserCode,string UserName,string UserPassword)

{

//初始化XML文档操作类

XmlDocument myDoc = new XmlDocument();

//加载XML文件

myDoc.Load(FileName);

//添加元素--UserCode

XmlElement ele = myDoc.CreateElement("UserCode");

XmlText text = myDoc.CreateTextNode(UserCode);

//添加元素--UserName

XmlElement ele1 = myDoc.CreateElement("UserName");

XmlText text1 = myDoc.CreateTextNode(UserName);

//添加元素--UserPwd

XmlElement ele2 = myDoc.CreateElement("UserPwd");

XmlText text2 = myDoc.CreateTextNode(UserPassword);

//添加节点 User要对应我们xml文件中的节点名字

XmlNode newElem = myDoc.CreateNode("element", "User", "");

//在节点中添加元素

newElem.AppendChild(ele);

newElem.LastChild.AppendChild(text);

newElem.AppendChild(ele1);

newElem.LastChild.AppendChild(text1);

newElem.AppendChild(ele2);

newElem.LastChild.AppendChild(text2);

//将节点添加到文档中

XmlElement root = myDoc.DocumentElement;

root.AppendChild(newElem);

//保存

myDoc.Save(FileName);

}

---------------------------删除----------------------------------------

//DeleteNode 完成对User的删除操作

//FileName 当前xml文件的存放位置

//UserCode 欲添加用户的编码

public void DeleteNode(string FileName, string UserCode)

{

//初始化XML文档操作类

XmlDocument myDoc = new XmlDocument();

//加载XML文件

myDoc.Load(FileName);

//搜索指定某列，一般是主键列

XmlNodeList myNode = myDoc.SelectNodes("//UserCode");

//判断是否有这个节点

if (!(myNode == null))

{

//遍历节点，找到符合条件的元素

foreach (XmlNode xn in myNode)

{

if (xn.InnerXml == UserCode)

//删除元素的父节点

xn.ParentNode.ParentNode.RemoveChild(xn.ParentNode);

}

}

//保存

myDoc.Save(FileName);

}

-----------------------------修改------------------------

//WriteXml 完成对User的修改密码操作

//FileName 当前xml文件的存放位置

//UserCode 欲操作用户的编码

//UserPassword 欲修改用户的密码

public void UpdateXML(string FileName, string UserCode, string UserPassword)

{

//初始化XML文档操作类

XmlDocument myDoc = new XmlDocument();

//加载XML文件

myDoc.Load(FileName);

//搜索指定的节点

System.Xml.XmlNodeList nodes = myDoc.SelectNodes("//User");

if (nodes != null)

{

foreach (System.Xml.XmlNode xn in nodes)

{

if (xn.SelectSingleNode("UserCode").InnerText == UserCode)

{

xn.SelectSingleNode("UserPwd").InnerText = UserPassword;

}

}

}

myDoc.Save(FileName);

}

# KeyCode和Keychar

if (e.Alt == true && e.KeyCode == Keys.G)//快捷键Alt+G

{

锁屏LToolStripMenuItem\_Click(null, null);

}

if (e.Control == true && e.KeyCode == Keys.M)//快捷键Control+M

{

NewForm();

}

e.KeyCode==Keys.Enter 和e.KeyChar==13和e.KeyChar===(Char)Keys.Enter 相等

e.KeyChar == (Char)48 ~ 57 -----> 0~9

e.KeyChar == (Char)8 -----------> Backpace

e.KeyChar == (Char)13-----------> Enter

//实例

private void txtName\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == 13)//判断是否按下Enter键

txtPwd.Focus();

e.Handled = true;//这个地方正规来说是要有的，它表示已经处理过KeyPress事件了

}

键盘 ASCII码 键盘 ASCII码

ESC 27 7 55

SPACE 32 8 56

! 33 9 57

" 34 : 58

# 35 ; 59

$ 36 < 60

% 37 = 61

& 38 > 62

' 39 ? 63

( 40 @ 64

) 41 A 65

\* 42 B 66

+ 43 C 67

' 44 D 68

- 45 E 69

. 46 F 70

/ 47 G 71

0 48 H 72

1 49 I 73

2 50 J 74

3 51 K 75

4 52 L 76

5 53 M 77

6 54 N 78

O 79 g 103

P 80 h 104

Q 81 i 105

R 82 j 106

S 83 k 107

T 84 l 108

U 85 m 109

V 86 n 110

W 87 o 111

X 88 p 112

Y 89 q 113

Z 90 r 114

[ 91 s 115

\ 92 t 116

] 93 u 117

^ 94 v 118

\_ 95 w 119

` 96 x 120

a 97 y 121

b 98 z 122

c 99 { 123

d 100 | 124

e 101 } 125

f 102 ~ 126

另外2个特殊ASCII码：回车=13，换行=10

# split用法

第一：static void Main(string[] args)

{

string aa = "1234,234523,4324,324";//清清月儿 http://blog.csdn.net/21aspnet/

string[] cc = aa.Split(new char[] { ',' }); 或者string[] sArray1=s.Split(new char[3]{‘c’,‘d’,‘e’}) ;

foreach (string bb in cc)

{

Console.Write(bb + "\r\n ");

}

}

第二：string aa = "1234,234523,4324,324";

string[] str = aa.Split(',');

foreach (string bb in str)

{

Console.Write(bb + " \r\n");

}

第三：static void Main(string[] args)

{

string a = "1,2,3,4,5,6,7,8,9";

string b = ","; 注意，第三种方法可以用字符串来切分

string[] c = Split(a, b);

foreach (string bb in c)

{

Console.Write(bb + " \r\n");

}

}

public static string[] Split(string input, string pattern)

{

string[] arr = System.Text.RegularExpressions.Regex.Split(input, pattern);

return arr;

}

# winform工具栏点击事件（两种方法）

//【第一中方法】

private void toolStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)//双击toolStrip后生成的click事件

{

switch (e.ClickedItem.Text)

{

case "统计":

{

if (comboBox1.Text == "用户缴费查询")

sum\_JiaoFei();

else if (comboBox1.Text == "用户取款查询")

sum\_QuKuan();

else if (comboBox1.Text == "缴费押金查询")

sum\_YaJin();

else if (comboBox1.Text == "押金取款查询")

sum\_QuYaJin();

break;

}

case "打印":

{

PrintDGV.Print\_DataGridView(dataGridView1, comboBox1.Text);

break;

}

case "退出":

{

this.Close();

break;

}

}

}

//【第二中方法】

private void toolStripButton8\_Click(object sender, EventArgs e)//双击toolStrip的某个子项后子项的click事件

{

}

# 正在运行，请稍后

if (td != null && td.IsAlive)

{

HDIC\_Message.ShowWarnDialog(null, "正在将数据导入到数据库，请等待……");

return;

}

td = new Thread(new ThreadStart(ImportData));// ImportData是你要运行的方法

td.SetApartmentState(ApartmentState.STA);

td.IsBackground = true;

td.Start();

# 测试当前网络是否畅通

/// <summary>

/// 测试服务器通信是否正常

/// </summary>

/// <param name="host">服务器IP地址</param>

/// <param name="port">服务器端口号， sql一般是1433</param>

/// <param name="millisecondsTimeout">超时</param>

/// <returns></returns>

public static bool TestConnection(string host, int port, int millisecondsTimeout)

{

var client = new System.Net.Sockets.TcpClient();

try

{

var ar = client.BeginConnect(host, port, null, null);

ar.AsyncWaitHandle.WaitOne(millisecondsTimeout);

return client.Connected;

}

catch

{

return false;

}

finally

{

client.Close();

}

}

# winfrom链接网址

System.Diagnostics.Process.Start("http://isline.cnblogs.com");

# 测试用时

System.Diagnostics.Stopwatch st = new System.Diagnostics.Stopwatch();

st.Start();

DataTable d1 = HDIC\_DB.GetList("select \* from chipdata where chipid='0000001111'");

st.Stop();

HDIC\_Message.ShowWarnDialog(null, "当前实例测量出总运行时间" + st.ElapsedMilliseconds + "毫秒");

# [C#中如何将字符串和byte[]互转](http://www.cnblogs.com/zhoufleru/archive/2007/06/04/770864.html)

## 字符串转换byte[]

第一种情况：

string   str   =   "abcd"  ;     
  byte[]   bytes   =   System.Text.Encoding.ASCII.GetBytes(str);

第二种情况：

private static byte[] HexStringToByteArray(string s)  
        {  
            s = s.Replace(" ", "");  
            byte[] buffer = new byte[s.Length / 2];  
            for (int i = 0; i < s.Length; i += 2)  
                buffer[i / 2] = (byte)Convert.ToByte(s.Substring(i, 2), 16);  
            return buffer;  
        }

## byte[]换成字符串

第一种情况：

byte[]   bytes   =   new   byte[255]   ;     
   string   str   =   System.Text.Encoding.ASCII.GetString(bytes,0,bytes.Length);

第二种情况：

  private string ByteArrayToHexString(byte[] data)  
        {  
            StringBuilder sb = new StringBuilder(data.Length \* 3);  
            foreach (byte b in data)  
                sb.Append(Convert.ToString(b, 16).PadLeft(2, '0').PadRight(3, ' '));  
            return sb.ToString().ToUpper();  
        }

或者

public static string byteToHexStr(byte[] bytes)

{

string returnStr = "";

if (bytes != null)

{

for (int i = 0; i < bytes.Length; i++)

{

returnStr += bytes[i].ToString("X2");

}

}

return returnStr;

}

## 字符串原型赋值到整型数组中

int[] CAIDBuffer = new int[11];

string CAID = "8007644278";

for (int i = 0; i < CAID.Length; i++)

{

CAIDBuffer[i] =Convert.ToInt32(CAID[i].ToString());

}

记住写法，如果是CAIDBuffer[i] =Convert.ToInt32(CAID[i]);则表示将char转换成int，这样写不是string转int 是错误的。

# DataGridView

## dataGridView1单元格选择值的问题

先设置datagridview的属性SelectionMode为FullRowSelect否则dataGridView1.SelectedRows[0].Cells[2].Value.ToString()和dataGridView1.SelectedCells[2].Value.ToString();不能用。

【

其他的如dataGridView1 .Rows[e.RowIndex ].Cells [0].Value可以用例如：

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

MessageBox.Show(dataGridView1 .Rows[e.RowIndex ].Cells [0].Value .ToString () );

}

】

第一：private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString()

和dataGridView1.SelectedRows[0].Cells[2].Value.ToString()

和Convert.ToString(dataGridView1[2, dataGridView1.CurrentCell.RowIndex].Value).Trim();

和dataGridView1.SelectedCells[2].Value.ToString();是相等的

}

第二：

private void 修改ToolStripMenuItem\_Click(object sender, EventArgs e)

{

只能用dataGridView1.SelectedRows[0].Cells[2].Value.ToString()

或者Convert.ToString(dataGridView1[2,dataGridView1.CurrentCell.RowIndex].Value).Trim();

或者dataGridView1.CurrentRow.Cells["roleName"].Value.ToString()

或者dataGridView1 .CurrentCell .Value.ToString()

或者dataGridView1.SelectedCells[2].Value.ToString();

不能用dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString()

}

## 获取选中的单元格信息

获取选中的dataGridView单元格个数

int32 selectedCellCount=dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

判断是否所有单元格全部选中

if(dataGridView1.AreAllCellSelected(true))

{MessageBox.Show("所有单元格都被选中了");

## dataGridView全选，反选

//全选

private void CheckAll\_CheckedChanged(object sender, EventArgs e)

{

if (dataGridView1.Rows.Count>0)

{

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

dataGridView1.Rows[i].Cells[0].Value = true;

}

}

}

【//另外，我们要实现一个CheckBox和DataGridView中所有的CheckBox的状态同步。DataGridView的名字为dgvRefObject，DataGridView的第一列（Column）为DataGridViewCheckBoxColumn；实现全选的CheckBox的名称为SelectAll。那么实现这个功能的代码如下：

dgvRefObject.DataSource = DataSet.Talbe[0];//重新初始化DataGridView

dgvRefObject.Columns[0].DefaultCellStyle.NullValue = SelectAll.Checked;】

//反选

private void CheckReverse\_CheckedChanged(object sender, EventArgs e)

{

if (dataGridView1.Rows.Count>0)

{

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if ((bool)dataGridView1.Rows[i].Cells[0].EditedFormattedValue)

{

dataGridView1.Rows[i].Cells[0].Value = false;

}

else

dataGridView1.Rows[i].Cells[0].Value = true;

}

}

}

## dataGridView按某列排序

【第一种方法：】

private void Form1\_Load(object sender, EventArgs e)

{

DataTable dt = new DataTable("ddd");

dt.Columns.Add(new DataColumn("a", typeof(int)));

dt.Columns.Add(new DataColumn("b", typeof(int)));

dt.Rows.Add(new object[] { "1", "1" });

dt.Rows.Add(new object[] { "3", "234" });

dt.Rows.Add(new object[] { "2", "32" });

dt.Rows.Add(new object[] { "23", "1" });

dt.Rows.Add(new object[] { "52", "231" });

dt.Rows.Add(new object[] { "34", "31" });

dt.Rows.Add(new object[] { "21", "51" });

dt.Rows.Add(new object[] { "2", "15" });

DataView dv = dt.DefaultView;

dv.Sort = "a desc";

DataGridView1.Source = dv;

}

【第二种：】

DataView dv = new DataView(dt, "", "a asc", DataViewRowState.CurrentRows);

## 添加自动编号

private void dgvDownloadList\_RowPostPaint(object sender, DataGridViewRowPostPaintEventArgs e)

{

Rectangle rectangle = new Rectangle(e.RowBounds.Location.X,

e.RowBounds.Location.Y,

dgvDownloadList.RowHeadersWidth - 4,

e.RowBounds.Height);

TextRenderer.DrawText(e.Graphics, (e.RowIndex + 1).ToString(),

dgvDownloadList.RowHeadersDefaultCellStyle.Font,

rectangle,

dgvDownloadList.RowHeadersDefaultCellStyle.ForeColor,

TextFormatFlags.VerticalCenter | TextFormatFlags.Right);

}

## 添加颜色

//给制定的单元格添加颜色

dataGridView1.Rows[stu\_name].Cells[i].Style.ForeColor=Color.Red;

//交替行显示

DataGridView控件的AlternatingRowsDefaultCellStyle属性实现交替行显示的效果，设置方法与RowDefaultCellStyle属性相同。还可以用RowTemplate（行模板）属性设置行的样式。

## 获取dgv中控件对象

if (e.Row.RowType == DataControlRowType.DataRow)

{

Label labKong = (Label)e.Row.FindControl("labKong1");

}

## 解决多线程后datagridview没有垂直滚动条

//用了多线程后datagridview没有垂直滚动条解决方法，有关具体实例可以查看自己写的“智能综合系统”的车辆信息模块

主要是CheckForIllegalCrossThreadCalls = false;

所以多线程直接控制UI造成的

后来改用委托

C# code

//定义公共变量

Thread te\_find;

delegate void DG\_Delegate();

delegate void DG\_label(string dgtext);

//在button1\_Click内使用多线程

te\_find = new Thread(new ThreadStart(hs\_find));

te\_find.Start();

//函数hs\_find调用UI的内容

{

this.BeginInvoke(new DG\_Delegate(ui\_dataGridView1)); //给dataGridView1赋查询到的数据

this.BeginInvoke(new DG\_label(ui\_label\_ta), "获得数据记录"); //修改label的Text

}

//两个被委托调用的函数

public void ui\_dataGridView1()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = hfds.Tables["psche"];

}

public void ui\_label\_ta(string text)

{

label\_ta.Text = text;

}

## DataGridView的绑定

DataGridView控件绑定到数据集，绑定方式有以下3种：

1）直接绑定到数据表

dataGridView.DataSource=ds.Tables[0];

2)绑定到数据表的数据视图：

dataGridView.DataSource=ds.Tables[0].DefaultView;

3)绑定到数据集：

dataGridView.DataSource=ds;

dataGridView.DataMember=ds.Tables[0].TableName;

dataGridView控件绑定到数据集以后仅仅复制了数据集的一个框架，还需要对数据集填充数据才能显示记录，调用DataAdapter对象的Fill方法为数据集填充数据。

# winform 主页模块调用方法

## 【方法一】

/// 每个窗体只允许打开一个

/// <typeparam name="T">窗体名称</typeparam>

/// <param name="mdiParent">父窗体</param>

private T OpenUniqueMDIChildWindow<T>(Form mdiParent) where T : Form, new()

{

foreach (Form subForm in mdiParent.MdiChildren)

{

if (subForm.GetType().Equals(typeof(T)))

{

subForm.Activate();

subForm.WindowState = FormWindowState.Normal;

return subForm as T;

}

}

T newForm = new T();

newForm.MdiParent = mdiParent;

newForm.Show();

newForm.WindowState = FormWindowState.Normal;

return newForm;

}

//例如模块一（门禁开卡锁卡）

private void toolStripButton10\_Click(object sender, EventArgs e)

{

OpenUniqueMDIChildWindow<baseinfo .car\_card .card >(this);

}

## 【方法二】

bool checkChildFrmExist(string childName)

{

foreach (Form childFrm in this.MdiChildren)

{

if (childFrm.Name == childName)

{

if (childFrm.WindowState == FormWindowState.Minimized)

childFrm.WindowState = FormWindowState.Normal;

childFrm.Activate();

return true;

}

}

return false;

}

//例如模块一（班级设置）

private void 班级设置MenuItem\_Click(object sender, EventArgs e)

{

if (!checkChildFrmExist("Class"))

{

Class frm = new Class();

frm.MdiParent = this;

frm.Show();

}

}

## 【方法三】

这个是haier 软件里面的。

private void expando\_ItemClick(object sender, EventArgs e)

{

string MenuName = ((TaskItem)sender).Name;

string MenuNo = ((TaskItem)sender).Tag.ToString();

if (string.IsNullOrEmpty(MenuName)) return;

//如果窗体已经创建，则将其激活，不再创建新事例

foreach (Form form in this.MdiChildren)

{

if (form.GetType().FullName.ToLower() == MenuName.ToLower())

{

form.Activate();

return;

}

}

Form f;

try

{

f = (Form)Activator.CreateInstance(Type.GetType(MenuName));

f.Text = ((TaskItem)sender).Text;

}

catch

{

return;

}

f.TopLevel = false;

f.MdiParent = this;

f.WindowState = FormWindowState.Maximized;

f.Show();

}

# Winform中Treeview绑定

## 非xml加载

private void bindTree() //

{

// 部门信息

SQL.SQL sql = new SQL.SQL();

DataTable dtBase\_Dpt = sql.GetData(sqlcon, "select SupDepBH,DepBH,DepName from DepTB where SupDepBH=0");

// 清空列表

treeDpt.Nodes.Clear();

if (dtBase\_Dpt.Rows.Count > 0)

{

// 添加根节点

treeDpt.Nodes.Add(dtBase\_Dpt.Rows[0]["DepBH"].ToString(), dtBase\_Dpt.Rows[0]["DepName"].ToString());

DataTable dtBase\_Dpt1 = sql.GetData(sqlcon, "select SupDepBH,DepBH,DepName from DepTB");

// 添加子节点

for (int i = 0; i < dtBase\_Dpt1.Rows.Count; i++)

{

if (dtBase\_Dpt1.Rows[i]["SupDepBH"].ToString() == dtBase\_Dpt.Rows[0]["DepBH"].ToString())

{

第一种方法： TreeNode tr = treeDpt.Nodes[0];//我测试发现Nodes[0]应该是0应该是深度的意思，不是第几个的意思

tr.Nodes.Add(dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), dtBase\_Dpt1.Rows[i]["DepName"].ToString());

AddTreeNode(dtBase\_Dpt1, dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), tr);

第二种方法：//｛treeDpt.Nodes [0].Nodes.Add(dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), dtBase\_Dpt1.Rows[i]["DepName"].ToString());

// AddTreeNode(dtBase\_Dpt1, dtBase\_Dpt1.Rows[i]["DepBH"].ToString(), treeDpt.Nodes [0]);｝

}

}

}

treeDpt.ExpandAll();

}

//递归调用

private void AddTreeNode(DataTable nodesData, string RootNode, TreeNode tmpNode)

{

// 定义当前节点

TreeNode tmpRootNode = tmpNode.Nodes[RootNode];

// 添加当前节点的子节点

for (int i = 0; i < nodesData.Rows.Count; i++)

{

// 添加当前节点的子接点

if (nodesData.Rows[i]["SupDepBH"].ToString() == RootNode)

{

tmpRootNode.Nodes.Add(nodesData.Rows[i]["DepBH"].ToString(),

nodesData.Rows[i]["DepName"].ToString());

AddTreeNode(nodesData, nodesData.Rows[i]["DepBH"].ToString(), tmpRootNode);

}

}

}

## 从xml中绑定：这个是长城自主开发

//主页面：

dtTemp = ds.Tables[DIRTREE];//this is Tree xml file

if (dtTemp == null) { printInfo = "Cannot find node,please try again..."; ConsoleU.writeLine(printInfo, ConsoleU.Level.Warning); return; }

if (dtTemp.Rows == null) { printInfo = "Cannot find node rows,please try again..."; ConsoleU.writeLine(printInfo, ConsoleU.Level.Warning); return; }

Int32 index = 0;

foreach (DataRow row in dtTemp.Rows)

{

List<String> tempnode = new List<String>();

if (Int32.Parse(row[PARENT].ToString()) == 0)

{

tempnode.Add(row[CURRENTNMAE].ToString());

tempnode.Add(row[CURRENT].ToString());

treeView\_dir.Nodes.Add(row[CURRENT].ToString(), row[CURRENTNMAE].ToString());

AddTreeNode(dtTemp.Rows, row[CURRENT].ToString(), treeView\_dir.Nodes[index++]);//这个index必须有，否则会错位

}

}

//下面是递归调用的方法：

private void AddTreeNode(DataRowCollection tempDtRows, string RootNodeID, TreeNode tmpNode)

{

Int32 index = 0;

if (tempDtRows == null) return;

// 添加当前节点的子节点

foreach (DataRow mrow in tempDtRows)

{

// 添加当前节点的子接点

if (mrow[PARENT].ToString() == RootNodeID)

{

string a = mrow[CURRENT].ToString();

string b = mrow[CURRENTNMAE].ToString();

tmpNode.Nodes.Add(a, b);

AddTreeNode(tempDtRows, mrow[CURRENT].ToString(), tmpNode.Nodes[index]);

index++;

}

}

}

//【添加图片：长城测试，先添加ImageList控件并往该控件添加图片，然后 treeView1.ImageIndex = 0;则第一级节点显示ImageList控件中第一个图片, treeView1.Nodes[0].ImageIndex = 2;则第二级节点显示第二个图片.

# Winform导出到excel

public void DataTableExportExcel(System.Data.DataTable ExportTable, params string[] HeadText)

{

try

{

Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.Interop.Excel.Application();//新的excel文档

excel.Application.Workbooks.Add(true); //新的工作表

excel.Visible = true;

int ColumnsIndex = ExportTable.Columns.Count;

//int RowsIndex = ExportTable.Rows.Count;

int index = 1;

//标题

for (; index <= ColumnsIndex; index++)

{

if (index <= HeadText.Length)

{

excel.Cells[1, index] = HeadText[index - 1];

}

}

//记录内容

for (int RowsIndex = 0; RowsIndex <= ExportTable.Rows.Count - 1; RowsIndex++)

{

for (index = 1; index <= ColumnsIndex; index++)

{

excel.Cells[RowsIndex + 2, index] = ExportTable.Rows[RowsIndex][index - 1].ToString();//行号

}

}

}

catch (Exception ex) { throw new Exception(ex.Message); }

}

private void btn\_daochu\_Click(object sender, EventArgs e)//调用excel，将信息导入到excel中

{

string[] headText = new string[7];

for (int ii = 0; ii < dgvClass.Columns.Count; ii++)

{

headText[ii] = dgvClass.Columns[ii].HeaderText;

}

DataTableExportExcel(dt, headText);

}

# winform和javascript互相调用

winform和javascript互相调用，借助winform的WebBrowser控件

//首先新建一个要调用的网页

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Main Page</title>

<script type="text/javascript" language="javascript">

function ShopXG(obj)

{

alert(obj);

}

function InvokeFunc()

{

window.external.ShowMessage("呵呵");

}

</script>

</head>

<body>

<input type="button" id="aa" value="测试" onclick="InvokeFunc();"/>

//该按钮目的是调用winform中对话框

</body>

</html>

//下面是建立Windows应用程序

1.创建Windows应用程序项目

2.在Form1窗体中添加WebBrowser控件

3.在Form1类的上方添加

using System.Security.Permissions;

namespace WindowsFormsApplication1

{

//要想调用JS的类都需要添加一下两句

[PermissionSet(SecurityAction.Demand, Name = "FullTrust")]

[System.Runtime.InteropServices.ComVisibleAttribute(true)]

//【这是为了将该类设置为com可访问。如果不进行该声明将会出错 】

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.webBrowser1.Url = new Uri(Application.StartupPath + @"\HTMLPage1.htm");

// this.webBrowser1.Navigate (Application.StartupPath + @"\HTMLPage1.htm");//和上面方法都可以

webBrowser1.ObjectForScripting = this; //【这句是必不可少的，是调用JS的前提作用是：将当前类设置为可由脚本访问】

}

//【注意：Application.StartupPath 长城测试必须有，否则报错，不知为什么。】

private void button1\_Click(object sender, EventArgs e)//目的调用javascript的对话框

{

webBrowser1.Document.InvokeScript("ShopXG",new object[]{"这是WinForm调用JS的一个测试！"});

//【webBrowser1.Document.InvokeScript("JS函数名", New Object() {参数})

参数值用:多个参数用分”，“号分开 。】

}

public void ShowMessage(string message)

{

MessageBox.Show("这是JS调用的WinFormc程序的一个测试！");

}

}

}

【详细分析：object[] paramArray=newobject[1];

paramArray[0]="g\_h";

webBrowser1.Document.InvokeScript("displayVar", paramArray);

也可以用这样来调用：

webBrowser1.Document.InvokeScript("JS函数名", New Object() {参数})

参数值用:多个参数用分”，“号分开 。

例如：

webBrowser1.Document.InvokeScript("Go", New Object() {"home","back"})

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

与脚本的互操作

在Case 1当中，我们已经看到，Web页面的HTML元素的事件，可以由Windows Form端来响应，可以在某种程度上看作是Web页面调用WinForm；那么反过

来，WinForm除了可以直接访问Web页面的HTML元素之外，能否调用Web页面里的各种Script呢？

首先是调用Web页面的脚本中已经定义好的函数。假设HTML中有如下Javascript：

function DoAdd(a, b) {

return a + b;

}

那么，我们要在WinForm调用它，只需如下代码即可：

object oSum = webBrowser.Document.InvokeScript("DoAdd", new object[] { 1, 2 });

int sum = Convert.ToInt32(oSum);

其次，如果我们想执行一段Web页面中原本没有的脚本，该怎么做呢？这次.Net的类没有提供，看来还要依靠COM了。IHTMLWindow2可以将任意的字符

串作为脚本代码来执行。

string scriptline01 = @"function ShowPageInfo() {";

string scriptline02 = @" var numLinks = document.links.length; ";

string scriptline03 = @" var numForms = document.forms.length; ";

string scriptline04 = @" var numImages = document.images.length; ";

string scriptline05 = @" var numScripts = document.scripts.length; ";

string scriptline06 = @" alert('网页的统计结果：\r\n链接数：' + numLinks + ";

string scriptline07 = @" '\r\n表单数：' + numForms + ";

string scriptline08 = @" '\r\n图像数：' + numImages + ";

string scriptline09 = @" '\r\n脚本数：' + numScripts);}";

string scriptline10 = @"ShowPageInfo();";

string strScript = scriptline01 + scriptline02 + scriptline03 + scriptline04 + scriptline05 +

scriptline06 + scriptline07 + scriptline08 + scriptline09 + scriptline10;

IHTMLWindow2 win = (IHTMLWindow2)webBrowser.Document.Window.DomWindow;

win.execScript(strScript, "Javascript")

# Winfrom鼠标移动窗体

【winfrom不用标题栏，鼠标移动窗体】

// 移动窗体

const int WM\_NCLBUTTONDOWN = 0xA1;

const int HT\_CAPTION = 0x2;

[DllImport("user32.dll")]

static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);

// 窗体上鼠标按下时

protected override void OnMouseDown(MouseEventArgs e)

{

if (e.Button == MouseButtons.Left & this.WindowState==FormWindowState.Normal)

{

// 移动窗体

this.Capture = false;

SendMessage(Handle, WM\_NCLBUTTONDOWN, HT\_CAPTION, 0);

}

}

# 委托

## 两个页面传值委托

//经查看多个网页，总结并实践通过。

详细委托，查看：

http://blog.csdn.net/ljsql/article/details/5421887

【第一：子窗体控制父窗体音乐】

子窗体代码：

namespace primaryTest.baseinfo

{

public delegate void pauseMedia();// 定义发布消息的委托

public partial class userrole : Form

{

public static event pauseMedia eventSend;//消息发布的事件

public userrole()

{

InitializeComponent();

}

private void tsbtn\_roleedit\_Click(object sender, EventArgs e)

{

eventSend();//这是与父窗体交互的委托事件

【注意：该触发事件必须在有(object sender, EventArgs e)的参数里面，否则会报错。。另外如果想让主窗体返回参数，可以这样定义： public delegate string pauseMedia();同时string a= eventSend();a就是主窗体返回的参数。】

}

}

}

父窗体代码：

private void 权限修改ToolStripMenuItem\_Click(object sender, EventArgs e)

{

webBrowser1.SendToBack();

//OpenUniqueMDIChildWindow<userrole>(this);

userrole userrole = new baseinfo.userrole();

userrole.MdiParent = this;

userrole .eventSend += new pauseMedia(eventSend);//这是委托，监听子窗体

userrole.Show();

}

private void eventSend()

{ axWindowsMediaPlayer1.Ctlcontrols.pause(); }

## Invoke的用法

说明：多线程编程中，我们经常要在工作线程中去更新界面显示，而在多线程中直接调用界面控件的方法是错误的做法，Invoke 和 BeginInvoke 就是为了解决这个问题而出现的，使你在多线程中安全的更新界面显示

直接在子线程中对窗体上的控件操作是会出现异常，这是由于子线程和运行窗体的线程是不同的空间，因此想要在子线程来操作窗体上的控件，是不可能简单的通过控件对象名来操作，但不是说不能进行操作，微软提供了Invoke的方法，其作用就是让子线程告诉窗体线程来完成相应的控件操作。

正确的做法是将工作线程中涉及更新界面的代码封装为一个方法，通过 Invoke 或者 BeginInvoke 去调用，两者的区别就是一个导致工作线程等待，而另外一个则不会。

using System.Threading;

//启动一个线程   
Thread thread=new Thread(new ThreadStart(DoWork));   
thread.Start();   
  
//线程方法   
private void DoWork()   
{   
this.TextBox1.Text="我是一个文本框";   
}   
  
如果你像上面操作,在VS2005或2008里是会有异常的...   
  
正确的做法是用Invoke\BeginInvoke  
  
using System.Threading;

namespace test

{

public partial class Form1 : Form

{

public delegate void MyInvoke(string str1);

public Form1()

{

InitializeComponent();

}

public void DoWork()

{

MyInvoke mi = new MyInvoke(UpdateForm);

this.BeginInvoke(mi, new Object[] {"我是文本框"});

}

public void UpdateForm(string param1)

{

this.textBox1.Text = param1;

}

private void button1\_Click(object sender, EventArgs e)

{

Thread thread = new Thread(new ThreadStart(DoWork));

thread.Start();

}

}

}

注意代理的使用

## 多个【radioButton】委托实现点击

private void jiajian\_Load(object sender, EventArgs e)

{

//下面是借助委托添加新事件，减轻了为每个radioButton 添加CheckedChanged事件

//this .radioButton1.CheckedChanged +=new EventHandler(radioButton1\_CheckedChanged);

//【上面radioButton1的一定要去掉了，因为 radioButton1\_CheckedChanged是它的，若再有则会执行两次。】

this.radioButton2 .CheckedChanged +=new EventHandler(radioButton1\_CheckedChanged);

this.radioButton3.CheckedChanged += new EventHandler(radioButton1\_CheckedChanged);

groupBox1.Visible = false;

pictureBox1.ImageLocation = a + @"\image\正在考试.gif";

}

//下面是事件方法：

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

if (!((RadioButton)sender).Checked)

{

return;

}

time = "";

switch (((RadioButton)sender).Tag.ToString())

{

case "10":

time = "10";

break;

case "20":

time = "20";

break;

case "30":

time = "30";

break;

default:

break;

}

}

# 多线程自己做的实例

private void Form1\_Load(object sender, EventArgs e)

{

Control.CheckForIllegalCrossThreadCalls = false;

/\*由于不允许在winform中直接跨线程访问控件的属性

加入这句代码以后发现程序可以正常运行了。

\* 这句代码就是说在这个类中我们不检查跨线程的调用是否合法（如果没有加这句话运行也没有异常，那么说明系统以及默认的采用了不检查的方式）。

\* 然而，这种方法不可取。我们查看CheckForIllegalCrossThreadCalls 这个属性的定义，就会发现它是一个static的，也就是说无论我们在项目的什么地方修改了这个值，

\* 他就会在全局起作用。而且像这种跨线程访问是否存在异常，我们通常都会去检查。如果项目中其他人修改了这个属性，那么我们的方案就失败了，我们要采取另外的方案。

\*/

}

Thread td1 = null;

private void btn\_seacher\_Click(object sender, EventArgs e)

{

if (td1 != null && td1.IsAlive)

{

MessageBox.Show("正在进行查询，请稍等……");

return;

}

td1 = new Thread(returndate);

td1.IsBackground = true;

td1.Start();

}

/// <summary>

/// 这个是查询的方法

/// </summary>

private void returndate()

{

try

{

SqlConnection con = new SqlConnection(" server=192.168.10.125;database=tcc;uid=sa;pwd=sa");

string sqlstr = " select \* from Car\_Data where 1=1";

if (dt1.Text.Trim() != "" && dt2.Text.Trim() != "")

{ sqlstr += " and '" + dt1.Text.Trim() + "'<DateTime and '" + dt2.Text.Trim() + "'>DateTime"; }

SqlCommand cmd = new SqlCommand(sqlstr, con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds);

DataTable dt = ds.Tables[0];

dataGridView1.DataSource = dt;

if (dt.Rows.Count > 0)

{

label1.Text = dt.Rows.Count.ToString();

}

//Thread.Sleep(1000);

}

catch { }

}

private void otherbutton\_Click(object sender, EventArgs e)

{

//另外一个按钮

if (td1 != null && td1.IsAlive)

{

MessageBox.Show("查询还没有结束，不能执行这个按钮功能……");

return;

}

else

MessageBox.Show("可以执行这个按钮功能了!!");

}

# 动态添加控件

public Form2()

{

InitializeComponent();

}

//用来添加控件的按钮

private void button1\_Click(object sender, EventArgs e)

{

TextBox txt = new TextBox();

txt.Location = new Point(150, 50);//位置

txt.Size = new Size(100, 20);//控件大小

txt.Name = "bt";//必须有，否则下面无法查看具体添加的的控件

this.Controls.Add(txt);

}

//用来显示新控件输入值的按钮

private void button2\_Click(object sender, EventArgs e)

{

TextBox tx = (TextBox)this.Controls.Find( "bt", true)[0];

button1.Text = tx.Text;

}

# 窗体之间显示顺序

## 控件成为顶级控件

private void 总在最前端ToolStripMenuItem\_Click(object sender, EventArgs e)

{

总在最前端ToolStripMenuItem.Checked = !总在最前端ToolStripMenuItem.Checked;

if (总在最前端ToolStripMenuItem.Checked)

{

this.TopMost = true;

this.Activate();

}

else

this.TopMost = false;

}

## 控件移到 Z 顺序的前面

btn\_ok.BringToFront();

## 控件移到 Z 顺序的后面

btn\_ok.SendToBack();

# ToolTip提示框

//将鼠标放到文本框上出现提示框：

private void txt\_name\_MouseEnter(object sender, EventArgs e)

{

toolTip1.Show("欢迎您点击姓名",txt\_name ,txt\_name .Width ,txt\_name .Height );

}

private void txt\_name\_MouseLeave(object sender, EventArgs e)

{

toolTip1.Hide(txt\_name );

}

# NotifyIcon控件

可以在窗体加载的时间添加如下代码：

notifyIcon1.Visible = true;

notifyIcon1.ShowBalloonTip(20,"通知","右击鼠标，可以进行更多功能操作",ToolTipIcon.Info);

//参数分别是（显示时间，标题，内容，图标类型）

//点击后主窗体最小化

private void notifyIcon1\_Click(object sender, EventArgs e)

{

if (this.WindowState == FormWindowState.Normal)

{

this.WindowState = FormWindowState.Minimized;

}

else if (this.WindowState == FormWindowState.Minimized)

{

this.WindowState = FormWindowState.Normal;

}

}

# DateTimePicker控件

使用 DateTimePicker 控件显示和选择时间 c# 的DateTimePicker 控件默认显示为“年月日”格式，当我们需要显示时间时可以将Format属性设置为"Time"或者自行设定具体的格式。

但是此时当我们点击右侧的小箭头时弹出的依然是选择日期，我们需要的是选择一个时间。怎么做呢？

只需要：将 DateTimePicker 的 ShowUpDown 属性设置为 true即可。

【对于自行设定格式，详解】首先将Format属性设置城Custom，然后将CustomFormat属性根据自己需要设置格式，如yy年-MM月-dd日 HH小时:mm分钟

:ss秒

# Listbox控件

## 判断新添加的信息在原列表中是否存在

（从(list\_left 列表右移到 list\_right表）

【第一种】 for (int i = 0; i < list\_left.SelectedItems.Count; i++)

{

if (list\_right.Items [i].ToString (). Equals(list\_left .SelectedItem.ToString()))

MessageBox.Show("已经存在！！");

else

list\_right.Items.Add(list\_left.SelectedItem.ToString());

}

【第二种】 if(list\_right.Items .Contains (list\_left .SelectedItem.ToString()))

MessageBox.Show("已经存在！！");

else

list\_right.Items.Add(list\_left.SelectedItem.ToString());

# OpenFileDialog控件

openFileDialog1.InitialDirectory = Application.StartupPath;

openFileDialog1.Filter = "音频（\*.mp3,\*.wma,\*wav）|\*.mp3;\*.wma;\*.wav|视频(\*.mp4,\*avi,\*.wmv)|\*.mp4;\*.avi;\*.wmv|所有文件(\*.\*)|\*.\*";

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

String fileExtension = System.IO.Path.GetExtension(openFileDialog1.FileNames[0]);

if (fileExtension != ".xml" || !System.IO.File.Exists(fileName))

{

ConsoleU.writeLine("this file is not xml or path error,please select again!", ConsoleU.Level.Warning);

}

}

# listView控件

## 乘法口诀

listView1.Items.Clear();

listView1.Columns.Add("", 60, HorizontalAlignment.Left);// 参数分别是指定文本、宽度、对齐方式

for (int i = 1; i < 10; i++)

{

listView1.Columns.Add(Convert.ToString(i), 60, HorizontalAlignment.Center );

}

for (int i = 1; i < 10; i++)

{

ListViewItem item = new ListViewItem(i.ToString());

for (int j = 1; j < 10; j++)

{

item.SubItems .Add(i.ToString() + "x" + j.ToString() + "=" + (i \* j).ToString());

}

listView1.Items.Add(item);//listView1添加项后自动换行

}

## View.Details

如果listView1.View = View.Details ;使用了这个视图必须要有标题否则不会显示内容.添加标题代码: listView1.Columns.Add("我是标题，可以添加很多标题", 60, HorizontalAlignment.Center );

## 显示图片

第一种方法：前台先放一个listview控件和imagelist控件

ListView a = (ListView)listView1;

ListViewItem b = new ListViewItem();

b.SubItems.Add("长城1");

b.SubItems.Add("长2");

a.Items.Add(b);

ImageList list = new ImageList();

list.Images.Add(Bitmap.FromFile (@"C:\Users\长城\Desktop\备份\ceshi\ceshi\resource\123.gif"));

a.SmallImageList = list;//用a.LargeImageList显示不出图片

b.ImageIndex = 0;//为imagelist里的图片索引值，这个必须有，否则图片不显示

this.Controls.Add(a);

第二种方法：

前台先放一个listview控件和imagelist控件

点击imagelist控件，在它的iamges属性中添加图片，点击listview控件将它的SmallImageList属性设置为imageList

ListView a = (ListView)listView1;//可以加到form \_load

ListViewItem lvi = new ListViewItem();

lvi.ImageIndex = x;//x为imagelist里的图片索引值

this.listView1.Items.Add(lvi );

# StreamReader

private void 打开ToolStripMenuItem\_Click(object sender, EventArgs e)

{

openFileDialog1.InitialDirectory = "d:\\";//定义初始目录

openFileDialog1.Filter = "文本文件(\*.txt)|\*.txt|观测数据文件(\*.\*\*o)|\*.\*\*o|导航文件(\*.\*\*n)|\*.\*\*n|气象数据文件(\*.\*\*m)" +

"|\*.\*\*m|GLONASS数据文件(\*.\*\*g)|\*.\*\*g|所有文件（\*.\*)|\*.\*";//过滤条件

if (this.openFileDialog1.ShowDialog() == DialogResult.OK)/\*C#中创建文件对话框的代码,

当点击文件对话框的确定按钮时打开相应的文件\*/

{

System.IO.StreamReader file = new System.IO.StreamReader(openFileDialog1.FileName);

#region 第一种，读取每一行

string readline;

while ((readline = file.ReadLine()) != null)

{

//对每一行进行处理

}

#endregion

#region 第二种，读取整个文本信息

this.file\_show.Text = file.ReadToEnd();

#endregion

}

}

代码设置控件图片

【1】panel3.BackgroundImage = Image.FromFile(Application .StartupPath+@"/image/fd1fe303e9fa033b3812bb30.jpg");

【2】if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.Image = new Bitmap(openFileDialog1.OpenFile());

}

【3】 if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.ImageLocation = openFileDialog1.FileName;

}

# StreamWriter

【提示：这是从一个小程序里面看到的,如果想看详细的还是搜寻下网上的吧.】

#region 添加日志

private void log(string message)

{

DateTime dt = DateTime.Now;

string fileName = Path.Combine(Application.StartupPath, "log.txt");

StreamWriter sw = new StreamWriter(fileName, true);

sw.WriteLine(message);

sw.WriteLine(string.Empty);

sw.Close();

}

# 门禁指纹（转十六进制）

门禁指纹“userid”（十进制）转换成“卡片号码”（十六进制） sql写法：

select right('00000000'+select right('00000000'+dbo.IntToHex(1196+131072),8)(1196+131072),8)

例如 ：

private void button1\_Click(object sender, EventArgs e)

{ //指纹转换

int a = Convert .ToInt32 ( textBox1.Text.Trim()) + 131072;

int b= Convert.ToInt32(textBox1.Text.Trim()) + 65536;

SqlConnection con = new SqlConnection("server=.;database=tcc;pwd=sa;uid=sa;");

con.Open();

SqlCommand com = new SqlCommand("select right('00000000'+dbo.IntToHex(" + a + "),8) as a , right('00000000'+dbo.IntToHex(" + b + "),8) as b", con);

SqlDataReader dr = com.ExecuteReader();

if (dr.Read())

{

string c = dr.GetString(0);

string d = dr.GetString(1).ToString();

MessageBox.Show("您转换成十六进制后两条分别是：" + c + "和" + d);

}

}

/\*数据库中没有系统自带的dbo.IntToHex方法：

可以在 新建查询 或者 标量值函数 中新建函数：\*/

//方法代码一：

create function [dbo].[hex](@cardno int )

returns varchar (100)

as

begin

declare @temp\_mod int

declare @i int

declare @result varchar(100)

declare @temp\_x int

declare @result\_values int

set @result=''

set @i=1

set @temp\_x=0

while @cardno>0

begin

set @temp\_mod=@cardno%16

set @cardno=@cardno/16

set @result=(case @temp\_mod when 10 then 'A'

when 11 then 'B'

when 12 then 'C'

when 13 then 'D'

when 14 then 'E'

when 15 then 'F'

else ltrim(str(@temp\_mod)) end )+@result end

return @result end

//方法代码二：

Create Function IntToHex(@IntNum int)

returns varchar(16)

as

begin

declare @Mods int,@res varchar(16)

set @res=''

while @IntNum <> 0

begin

set @Mods =@IntNum % 16

if @Mods > 9

set @res = Char(Ascii('A')+@Mods-10)+@res

else

set @res = Cast(@Mods as varchar(4)) + @res

set @IntNum = @IntNum/16 end

return @res end

# 苑斌做的用户权限

第一：

这个是添加用户权限（用temStr 获取选中的(CheckBox) 最后把这个字符串放入数据表中）

private void btn\_addRole\_Click(object sender, EventArgs e)

{

string temStr = "";

foreach (Control c in this.Controls)

{

if (c.GetType() == typeof(CheckBox) && (c as CheckBox).Checked)

{

temStr += c.TabIndex.ToString() + ",";//这个地方用Tab键获取你选中了哪个

}

}

【我用的方法是：foreach (Control c in groupbox1.Controls)

{

if (c is CheckBox)

{ if ((c as CheckBox).Checked) { (c as CheckBox).Checked = false; } }

}

】

if (!temStr.Equals(""))

{

temStr = temStr.Substring(0, temStr.Length - 1);

}

this.DialogResult = DialogResult.OK;

}

//数据备份0//数据恢复1//用户管理2//角色管理3//设备管理4//人员管理5//线路管理6//排班管理7//人员分配8//查询统计9//统计报表10//参数设置11//指纹注册参数设置12//命令结果检测13//修改终端编号14//指纹管理15//节假日管理16//请假管理17

第二：

主页面如数据查询模块：

private void 数据查询ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Boolean tempStr = false;

tempStr = permResult("frmShowDetail");//调用方法用于判断权限

if (tempStr == true)

{

OpenUniqueMDIChildWindow<frmShowDetail>(this);

}

else

{

MessageBox.Show("您没有相关访问权限，请与管理员联系！");

}

}

权限方法：

private Boolean permResult(string winName)

{

Boolean result = false;

if (PubVar.tempUsername != "admin")

{

string tempStr = string.Empty;

string tempRole = string.Empty;

switch (winName)

{

case "beifen": tempStr = "0"; break;

case "RestoreData": tempStr = "1"; break;

case "user": tempStr = "2"; break;

case "RoleMan": tempStr = "3"; break;

case "EquipmentMan": tempStr = "4"; break;

case "PersonMan": tempStr = "5"; break;

case "frmLineManager": tempStr = "6"; break;

case "frmArrangeClass": tempStr = "7"; break;

case "frmPersonToClass": tempStr = "8"; break;

case "frmShowDetail": tempStr = "9"; break;

case "ParameterSet": tempStr = "11"; break;

case "CommFrm": tempStr = "12"; break;

case "commandRequest": tempStr = "13"; break;

case "frmUpdateID": tempStr = "14"; break;

case "FingerprintIssued": tempStr = "15"; break;

case "frmHoliday": tempStr = "16"; break;

case "frmVacation": tempStr = "17"; break;

}

tempRole = PubVar.temproleList;

string[] str = tempRole.Split(',');

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

{

if (tempStr == str[i])

{

result = true;

break;

}

else

{

result = false;

}

}

}

else

{ result = true; }

return result;

}

第三：

登录界面如果登录成功则：

if (ds.Tables[0].Rows.Count != 0)

{

PubVar.temproleList = ds.Tables[0].Rows[0][1].ToString();//将该人员权限字段值赋值给公共的变量

}

# Winform特效

## 窗体飘动

public partial class Form1 : Form

{

int ScreenWidth = SystemInformation.PrimaryMonitorMaximizedWindowSize.Width;

int ScreenHeight = SystemInformation.PrimaryMonitorMaximizedWindowSize.Height;

public Form1()

{

InitializeComponent();

}

//控制水平移动

private void timer1\_Tick(object sender, EventArgs e)

{

Point p = new Point(this.DesktopLocation.X, this.DesktopLocation.Y);

if (p.X + Width < ScreenWidth)

{

this.DesktopLocation = new Point(p.X + 1, p.Y);

}

else

{

this.DesktopLocation = new Point(0, 0);

}

}

//控制垂直移动

private void timer2\_Tick(object sender, EventArgs e)

{

Point p = new Point(this.DesktopLocation.X, this.DesktopLocation.Y);

if (p.Y + Height < ScreenHeight)

{

this.DesktopLocation = new Point(p.X, p.Y + 1);

}

else

{

this.DesktopLocation = new Point(0, 0);

}

}

//控制漂移运动

private void timer3\_Tick(object sender, EventArgs e)

{

Point p = new Point(this.DesktopLocation.X, this.DesktopLocation.Y);

if (p.X + Width < ScreenWidth || p.Y + Height < ScreenHeight)

{

this.DesktopLocation = new Point(p.X + 1, p.Y + 1);

}

else

{

this.DesktopLocation = new Point(0, 0);

}

}

private void btnH\_Click(object sender, EventArgs e)

{

this.timer1.Enabled = true;

this.timer2.Enabled = false;

this.timer3.Enabled = false;

}

private void btnV\_Click(object sender, EventArgs e)

{

this.timer1.Enabled = false;

this.timer2.Enabled = true;

this.timer3.Enabled = false;

}

private void btnDrift\_Click(object sender, EventArgs e)

{

this.timer1.Enabled = false;

this.timer2.Enabled = false;

this.timer3.Enabled = true;

}

private void btnStop\_Click(object sender, EventArgs e)

{

this.timer1.Enabled = false;

this.timer2.Enabled = false;

this.timer3.Enabled = false;

}

}

## 汉字飘动

public partial class Form2 : Form

{

int ScreenWidth = SystemInformation.PrimaryMonitorMaximizedWindowSize.Width;

int ScreenHeight = SystemInformation.PrimaryMonitorMaximizedWindowSize.Height;

public Form2()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

this.timer1.Interval = 10;//时º¡À间?间?隔?为a10

this.timer1.Enabled = true;

this.WindowState = FormWindowState.Maximized;////窗ä¡ã口¨²最Á?大ä¨®化¡¥

}

private void timer1\_Tick(object sender, EventArgs e)

{

label1.Left = label1.Left - 3;

if (label1.Right < 0)

{

label1.Left = this.Width;

}

}

}

## 窗体淡入淡出

窗体淡入，（点击form1按钮后form2淡入显示。现将form2的Opacity属性设置为0%）淡出方法和淡入相似

Form2 a = new Form2();

public Form1()

{

InitializeComponent();

timer1.Enabled = false;

}

private void button1\_Click(object sender, EventArgs e)

{

a.Show();

timer1.Enabled = true;

this.Hide();

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (a.Opacity <1)

{

a.Opacity = a.Opacity + 0.1;

}

else if (a.Opacity > 1)

{timer1.Enabled = false;

}

}

}

# 附加知识

## 支持gif格式的控件

pictureBox1和label都可以。

## winform背景音乐

给winform添加背景音乐：只支持wav格式

using System.Media;

SoundPlayer msc = new SoundPlayer(@"C:\Documents and Settings\Administrator\桌面\恋爱广播录音1.WAV");

msc.Play();

## winform语音

引用DotNetSpeech组件，功能均来自此组件。

提示：

步骤一:先写下面的代码

SpeechVoiceSpeakFlags SpFlags = SpeechVoiceSpeakFlags.SVSFlagsAsync;

SpVoice sp = new SpVoice();

sp.Voice = sp.GetVoices("name=Microsoft Simplified Chinese","").Item(0); //用中文男声读,女声是用microsoft Mary,要去下载

sp.Voice = sp.GetVoices("name=Microsoft Sam", "").Item(0); //用英文读

sp.Rate = -3;

sp.Speak(this.textBox1.Text, SpFlags);

步骤二:接下来,还要去控制面板配置Speech SDK引擎当前所处理语言。打开"控制面板"，打开"语音"配置项目，可以看到在这里我们可以指定当前可以对何种语言进行识别或者合成，

还可以配置相关的硬件设备以及控制语速,在"文字-语音转换"的"语音选择"组合框中选择简体中文(Microsoft Simplified Chinese)。这样就可以合成汉语文字了。

如果没有Microsoft Simplified Chinese这项,就去网上找一个

## 对一个方法想延迟几秒钟再执行

一个方法中如果想延迟几秒钟在执行本方法，不用将 System.Threading.Thread.Sleep(2000);放在本方法中，因为放进去他要等该方法执行完后才延迟。我的解决方法：可以用时间控件time，设置一个变量，时间控件根据变量值执行该方法下一步。如：

private void timer2\_Tick(object sender, EventArgs e)

{

if (Convert.ToBolean(beginrun))//自动出题

{

beginrun = 0;

fanhui(Convert.ToInt32(txt\_min.Text.Trim()), Convert.ToInt32(txt\_max.Text.Trim()));//出题

lbl\_result.Text = "答案显示";

}

beginrun--;

}

int beginrun=0;

private void btn\_ok()

{

time2.Enable=true;

time2.Interval=1000;

begirnun=2;

}

## 串口方面的知识

一般从串口接收到数据后，如果想将多个字节拼接，一般定义一个无符号的Uint32类型变量（晓淯说的）

## 大端和小端

Little Endian还是Big Endian与操作系统和芯片类型都有关系。

vc6.0 里默认是按小端格式输出，这是因为X86结构是小端模式。而KEIL C51则为大端模式。很多的ARM，DSP都为小端模式。有些ARM处理器还可以由硬件来选择是大端模式还是小端模式。  
    大端：PowerPC、MAC OS、KEIL C51  
    小端：x86、ARM、DSP、vc6.0  
    大端：低地址存放高字节数据，高地址存放低字节  
    小端：低地址存放低字节数据，高地址存高低字节

为什么会有大小端模式之分呢？  
    这是因为在计算机系统中，我们是以字节为单位的，每个地址单元都对应着一个字节，一个字节为8bit。但是在C语言中除了8bit的char之外，还有16bit的short型，32bit的long型（要看具体的编译器），另外，对于位数大于8位的处理器，例如16位或者32位的处理器，由于寄存器宽度大于一个字节，那么必然存在着一个如何将多个字节存储的问题。因此就导致了大端存储模式和小端存储模式。例如一个16bit的short型x，在内存中的地址为0x0010，x的值为0x1122，那么0x11为高字节(最高字节MSB（Most Significant Byte）)，0x22为低字节(最低字节LSB（Least Significant Byte）)。对于大端模式，就将0x11放在低地址中，即0x0010中，0x22放在高地址中，即0x0011中。小端模式，刚好相反。

#### 面这段代码可以用来测试一下你的编译器是大端模式还是小端模式：

#### short int x; char x0,x1; x=0x1122; x0=((char\*)&x)[0];  //低地址单元 x1=((char\*)&x)[1];  //高地址单元

#### printf("x0=%x x1=%x\n",x0,x1);

#### 如果x0=11表示是大端；如果x0=22则表示小端