# 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);

}

}

# 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);

}

# 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()

}

# 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");

}

# 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;

}

# 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;

}

}

}

//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);

# 公共类

/// <summary>

/// 取得本机串口

/// </summary>

/// <returns></returns>

public static string[] GetHostCOM()

{

string[] sAllPort = null;

try

{

sAllPort = SerialPort.GetPortNames();

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

return sAllPort;

}

/// <summary>

/// 获取本地IP地址

/// </summary>

/// <returns></returns>

public static string GetHostIPAddress()

{

string strHostIPAddr;

strHostIPAddr = "";

System.Net.IPAddress[] addressList = Dns.GetHostEntry(Dns.GetHostName()).AddressList;

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

//{

// strHostIPAddr += addressList[i].ToString();

//}

strHostIPAddr += addressList[addressList.Length - 1].ToString();

return strHostIPAddr;

}

/// <summary>

/// 判断对象是否是整型数

/// </summary>

/// <param name="obj"></param>

/// <returns></returns>

public static bool CheckObjectIsInteger(object obj)

{

obj = obj.ToString().Trim();

if (string.IsNullOrEmpty(obj.ToString()))

{

return false;

}

else

{

return Regex.IsMatch(obj.ToString(), @"^\d{1,}$");

}

}

/// <summary>

/// 验证对象是否是字节数组

/// </summary>

/// <param name="objValue"></param>

/// <returns></returns>

public static bool CheckObjIsByteArray(object objValue)

{

if (objValue is byte[])

{

return true;

}

else

{

return false;

}

}

/// <summary>

/// 验证身份证号码的格式

/// </summary>

/// <param name="IDNo"></param>

/// <returns></returns>

public static bool IsValidIdentityNo(string IdentityNo)

{

IdentityNo = IdentityNo.Trim();

if (string.IsNullOrEmpty(IdentityNo))

{

return false;

}

else

{

return Regex.IsMatch(IdentityNo, @"^(\d{14}(\d{1}|\d{4}|\d{3}[xy]))$");

}

}

/// <summary>

/// 防注入方法

/// </summary>

/// <param name="text"></param>

/// <returns></returns>

public static string InputText(string text)

{

text = text.Trim();

if (string.IsNullOrEmpty(text))

return string.Empty;

// 替换两个或两个以上的空格为一个空格

text = Regex.Replace(text, "[\\s]{2,}", " ");

// 替换<br>为转义字符“\n”

text = Regex.Replace(text, "(<[b|B][r|R]/\*>)+|(<[p|P](.|\\n)\*?>)", "\n");

// 替换"&nbsp;"为空格

text = Regex.Replace(text, "(\\s\*&[n|N][b|B][s|S][p|P];\\s\*)+", " ");

// 替换特殊字符

text = Regex.Replace(text, "<(.|\\n)\*?>", string.Empty);

// 替换单引号

text = text.Replace("'", "''");

return text;

}

/// <summary>

/// 生成指定个数的随机数

/// </summary>

/// <param name="codeCount"></param>

/// <returns></returns>

private string CreateRandomCode(int codeCount)

{

string allChar = "0,1,2,3,4,5,6,7,8,9";

string[] allCharArray = allChar.Split(',');

string randomCode = "";

int temp = -1;

Random rand = new Random();

for (int i = 0; i < codeCount; i++)

{

if (temp != -1)

{

rand = new Random(i \* temp \* ((int)DateTime.Now.Ticks));

}

int t = rand.Next(10);

if (temp == t)

{

return CreateRandomCode(codeCount);

}

temp = t;

randomCode += allCharArray[t];

}

return randomCode;

}

/// <summary>

/// 生成20位的主键数，如AB200807021438250001

/// </summary>

/// <param name="str">主键字符串的前两位</param>

/// <param name="str">例如调用接口：PKey.CreateKeyStr("RO");返回RO打头的20位字符</param>

/// <returns></returns>

public static string CreateKeyStr(string str)

{

StringBuilder strKey = new StringBuilder();

Random rd = new Random(System.DateTime.Now.Millisecond);

string strDate;

string strRd;

str = str.ToUpper().Replace(" ", "") + "AA";

str = str.Substring(0, 2);

strKey.Append(str);

strDate = DateTime.Now.ToString("yyyyMMdd") + DateTime.Now.ToString("HH:mm:ss").Replace(":", "");

strKey.Append(strDate);

strRd = "0000" + rd.Next(1, 10000);

strRd = strRd.Substring(strRd.Length - 4, 4);

strKey.Append(strRd);

return strKey.ToString();

}

/// <summary>

/// 系统重启

/// </summary>

public static void Reset()

{

Application.ExitThread();

System.Threading.Thread thtmp = new System.Threading.Thread(new System.Threading.ParameterizedThreadStart(run));

object appName = Application.ExecutablePath;

//System.Threading.Thread.Sleep(2000);

thtmp.Start(appName);

}

private static void run(Object obj)

{

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

ps.StartInfo.FileName = obj.ToString();

ps.Start();

}

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

/// <summary>

/// 导Ì?出?DataGridView数ºy据Y 长¡è城?修T改?

/// </summary>

/// <param name="dgv">DataGridView控?件t</param>

public void outPutDataGridViewData(DataGridView dgv, string txtTitle)

{

SaveFileDialog save = new SaveFileDialog();

//save.Filter = "user files(" + filter + ")|" + filter";

save.Filter = "文?本À?格?式º?.txt|.txt|表À¨ª格?.xls|.xls|文?档Ì¦Ì.doc|.doc|RLC格?式º?.rlc|.rlc";

save.Title = "导Ì?出?文?件t到Ì?";

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

{

string padStr = "";//用®?于®¨²填¬?充?空?格?

Stream myStream = save.OpenFile();

StreamWriter sw = new StreamWriter(myStream, System.Text.Encoding.GetEncoding("GB2312"));

try

{

int[] temp = new int[dgv.Columns.Count];

//for (int index = 0; index < dgv.Columns.Count; index++)

for (int index = 1; index < dgv.Columns.Count; index++)

{

temp[index] = GetMaxWidth(dgv, index);

}

//写¡ä标À¨º题¬a

sw.WriteLine(txtTitle);

//写¡ä数ºy据Y字Á?段?

string tempTitle = "";

//for (int i = 0; i < dgv.Columns.Count; i++)

for (int i = 1; i < dgv.Columns.Count; i++)

{

if (save.FilterIndex == 4)//如¨?果?是º?表À¨ª格?xls格?式º?

{

//if (i > 0)

if (i > 1)

{

tempTitle += "\t";

}

tempTitle += dgv.Columns[i].Name;

}

else//如¨?果?是º?doc、¡étxt等Ì¨¨格?式º?

{

//if (i == 0)

if (i == 1)

{

tempTitle += dgv.Columns[i].Name;

}

else

{

if (temp[i - 1] - Encoding.Default.GetBytes(dgv.Columns[i - 1].Name).Length < 0)

{

tempTitle += padStr.PadRight(2, ' ');

}

else

{

tempTitle += padStr.PadRight(temp[i - 1] - Encoding.Default.GetBytes(dgv.Columns[i - 1].Name).Length + 2, ' ');

}

tempTitle += dgv.Columns[i].Name;

}

}

}

sw.WriteLine(tempTitle);

//循-环¡¤写¡ä内¨²容¨Y

for (int j = 0; j < dgv.Rows.Count; j++)

{

string tempStr = "";

//for (int k = 0; k < dgv.Columns.Count; k++)

for (int k = 1; k < dgv.Columns.Count; k++)

{

if (save.FilterIndex == 4)//如¨?果?是º?表À¨ª格?xls格?式º?

{

//if (k > 0)

if (k > 1)

{

tempStr += "\t";

}

tempStr += "'" + dgv.Rows[j].Cells[k].Value.ToString().Trim();

}

else //如¨?果?是º?doc、¡étxt等Ì¨¨格?式º?

{

//if (k == 0)

if (k == 1)

{

tempStr += dgv.Rows[j].Cells[k].Value.ToString();

}

else

{

if (temp[k - 1] - Encoding.Default.GetBytes(dgv.Columns[k - 1].Name).Length < 0)

{

tempStr += padStr.PadRight(Encoding.Default.GetBytes(dgv.Columns[k - 1].Name).Length - Encoding.Default.GetBytes(dgv.Rows[j].Cells[k - 1].Value.ToString().Trim()).Length + 2, ' ');

}

else

{

tempStr += padStr.PadRight(temp[k - 1] - Encoding.Default.GetBytes(dgv.Rows[j].Cells[k - 1].Value.ToString().Trim()).Length + 2, ' ');

}

tempStr += dgv.Rows[j].Cells[k].Value.ToString().Trim();

}

}

}

sw.WriteLine(tempStr);

}

MessageBox.Show("导Ì?出?数ºy据Y成¨¦功|！ê?", "提¬¨¢示º?", MessageBoxButtons.OK, MessageBoxIcon.Information);

sw.Close();

myStream.Close();

}

catch (Exception ee)

{

MessageBox.Show(ee.Message);

}

finally

{

sw.Close();

myStream.Close();

}

}

}

/// 配合上面方法得到DataGridView每列最大的宽度

/// </summary>

/// <returns></returns>

private int GetMaxWidth(DataGridView dgv, int columnIndex)

{

int max = 0;

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

{

max = max > Encoding.Default.GetBytes(dgv.Rows[i].Cells[columnIndex].Value.ToString().Trim()).Length ? max : Encoding.Default.GetBytes(dgv.Rows[i].Cells[columnIndex].Value.ToString().Trim()).Length;

}

return max;

}

}