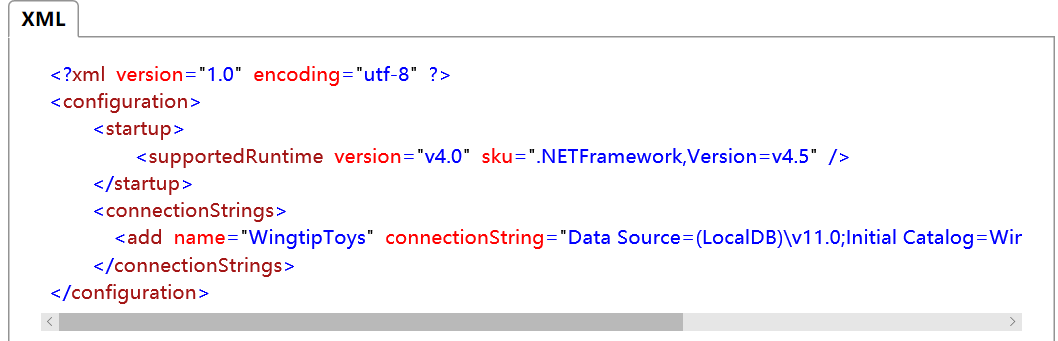
The T:System.Configuration.ConfigurationManager class enables you to access machine, application, and user configuration information.

This class replaces the T:System.Configuration.ConfigurationSettings class, which is deprecated.

ConnectionStrings节点下的节点组成集合。name作为key。每个节点的属性用点访问

var connectionString = ConfigurationManager.ConnectionStrings["WingtipToys"].ConnectionString;



Add Custom Configuration Elements In .NET

<http://www.c-sharpcorner.com/article/add-custom-configuration-elements-in-net/>

建立内部标签ConfigurationElement

建立标签集合CompanyElementCollection

建立包含集合的section ConfigurationSection

建立组

our Company Element is ready. We need to create a collection of type Company Element

If we don’t supply the AddItemName, we will get add tag instead of company tag.

1. [ConfigurationCollection(**typeof**(CompanyElement), AddItemName = "company")]
2. **public** **class** CompanyElementCollection : ConfigurationElementCollection
3. {
4. **protected** override ConfigurationElement CreateNewElement()
5. {
6. **return** **new** CompanyElement();
7. }
9. **protected** override object GetElementKey(ConfigurationElement element)
10. {
11. **return** ((CompanyElement)element).Name;
12. }
14. }

<companies>

<company name="Microsoft Corporation" shortName="MSFT" companyCode="MSFT"/>

<company name="Yahoo" shortName="YHOO" companyCode="YHOO"/>

</companies>

<protocols>

<add name="HttpSoap" />

<add name="HttpPost" />

<add name="HttpGet" />

<add name="Documentation" />

</protocols>

Now, we will create a new class MySectionGroup that inherits from ConfigurationSectionGroup. In this class, we will add two public properties that will return the sections, which we have created.

1. #region MySection Group
2. **public** **class** MySectionGroup : ConfigurationSectionGroup
3. {
4. [ConfigurationProperty("setting", IsRequired = **false**)]
5. **public** SettingSection GeneralSettings
6. {
7. **get** { **return** (SettingSection)**base**.Sections["setting"]; }
8. }
10. [ConfigurationProperty("companySection", IsRequired = **false**)]
11. **public** CompanySection ContextSettings
12. {
13. **get** { **return** (CompanySection)**base**.Sections["companySection"]; }
14. }
15. }
16. #endregion

<configSections>

<sectionGroup name="mysection" type="CustomTags.MySectionGroup,CustomTags">

<section name="companySection" type="CustomTags.CompanySection,CustomTags"/>

<section name="settingSection" type="CustomTags.SettingSection,CustomTags"/>

</sectionGroup>

</configSections>

1. <mysection>
2. <settingSection countrycode="US" isenabled='true' />
3. <companySection>
4. <companies>
5. <company name="Microsoft Corporation" shortName="MSFT" companyCode="MSFT"/>
6. <company name="Yahoo" shortName="YHOO" companyCode="YHOO"/>
7. </companies>
8. </companySection>
9. </mysection>

Why, Where, and How of .NET Configuration Files

<https://www.codeproject.com/Articles/616065/Why-Where-and-How-of-NET-Configuration-Files>

Machine Configuration

Application Configuration

User Settings

Other Configuration

If this isn't all confusing enough, you should be aware of a few other files. There is a root Web.config (located in the same directory as machine.config). Also, sub-directories of a web application may provide additional overrides of inherited settings, via a Web.config specific to that sub-directory.

Lastly, IIS provides some of its own configuration. A typical location would be: C:\Windows\System32\inetsrv\ApplicationHost.config

With .NET 2.0, Microsoft tried to make it even easier to use configuration files. They introduced a settings file. A careful observer will note that the "settings" start their life in the application configuration file and, later, get copied to the user settings configuration file.

With Windows Form and WPF applications, you'll find a file Settings.settings in the Properties folder of your project.

PerUserRoaming PerUserRoamingAndLocal？

通过使用键盘 （选项卡上，按 SHIFT + TAB 等），通过调用更改焦点时 [Select](https://msdn.microsoft.com/zh-cn/library/7wt11hea.aspx)或 [SelectNextControl](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.selectnextcontrol.aspx) 方法，或通过设置 [ContainerControl.ActiveControl](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.containercontrol.activecontrol.aspx) 属性设置为当前窗体，焦点事件按下列顺序发生︰

1. [Enter](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.enter.aspx)
2. [GotFocus](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.gotfocus.aspx)
3. [Leave](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.leave.aspx)
4. Validating
5. [Validated](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.validated.aspx)
6. [LostFocus](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.lostfocus.aspx)

当更改焦点时通过使用鼠标或调用 [Focus](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.focus.aspx) 方法时，按以下顺序执行的焦点事件︰

1. [Enter](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.enter.aspx)
2. [GotFocus](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.gotfocus.aspx)
3. [LostFocus](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.lostfocus.aspx)
4. [Leave](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.leave.aspx)
5. Validating
6. [Validated](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.validated.aspx)

如果 [CausesValidation](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.causesvalidation.aspx) 属性设置为 **false**, 、 Validating 和 [Validated](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.validated.aspx) 所抑制的事件。

如果 [Cancel](https://msdn.microsoft.com/zh-cn/library/system.componentmodel.canceleventargs.cancel.aspx) 属性 [CancelEventArgs](https://msdn.microsoft.com/zh-cn/library/system.componentmodel.canceleventargs.aspx) 设置为 **true** 中 Validating 事件委托中，通常会发生后的所有事件 Validating 事件，将会抑制。

请不要尝试将焦点设置在 [Enter](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.enter.aspx), ，[GotFocus](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.gotfocus.aspx), ，[Leave](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.leave.aspx), ，[LostFocus](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.lostfocus.aspx), ，Validating, ，或 [Validated](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.validated.aspx) 事件处理程序。 这样做可能导致您的应用程序或操作系统停止响应。

String.IsNullOrEmpty 可以判断

string a = null;

string b = string.Empty;

string c = "";

String.IsNullOrWhiteSpace 可以判断

string a = null;

string b = string.Empty;

string c = "";

string d = " ";



输出

A逻辑

B逻辑

C逻辑

Multi Threading With Windows Forms

<http://www.c-sharpcorner.com/article/multi-threading-with-windows-forms/>

The reason I am using a ThreadPool object is that creating threads is a very expensive operation. Therefore, using the pool means that on completion, threads can be put into a pool to be reused.

//queue work so a thread from the thread pool can pick it up and execute it

ThreadPool.QueueUserWorkItem((o) => this.PerformWork(ARec));

Convert Text to Speech

“Add References”. Click on Framework tab, and then add "System.Speech".

Add the following namespace: using System.Speech.Synthesis;

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

{

SpeechSynthesizer speechSynthesizer = new SpeechSynthesizer();

speechSynthesizer.Volume = 100; // 0...100

speechSynthesizer.Rate = 0; // -10...10

// Synchronous - Speaks the specified text string.

// speechSynthesizer.Speak(textBox1.Text);

// Asynchronous -   Speaks the specified text string asynchronously.

speechSynthesizer.SpeakAsync(textBox1.Text);

}

Steps To Add Combobox Inside Datagridview Window Form

<http://www.c-sharpcorner.com/blogs/steps-to-add-combobox-inside-datagridview-window-form>

Add combo box in Datagridview cell on cellClick event

1. **private** **void** dgv\_CellClick(**object** sender, DataGridViewCellEventArgs e)
2. {
3. **if** (e.ColumnIndex > -1)
4. {
5. // Bind grid cell with combobox and than bind combobox with datasource.
6. DataGridViewComboBoxCell l\_objGridDropbox = **new** DataGridViewComboBoxCell();
8. // Check the column  cell, in which it click.
9. **if** (dgv.Columns[e.ColumnIndex].Name.Contains("Description"))
10. {
11. // On click of datagridview cell, attched combobox with this click cell of datagridview
12. dgv[e.ColumnIndex, e.RowIndex] = l\_objGridDropbox;
13. l\_objGridDropbox.DataSource = GetDescriptionTable(); // Bind combobox with datasource.
14. l\_objGridDropbox.ValueMember = "Description";
15. l\_objGridDropbox.DisplayMember = "Description";
17. }
19. **if** (dgv.Columns[e.ColumnIndex].Name.Contains("PaidWith"))
20. {
21. dgv[e.ColumnIndex, e.RowIndex] = l\_objGridDropbox;
22. l\_objGridDropbox.DataSource = GetPaidWithTable();
23. l\_objGridDropbox.ValueMember = "PaidWith";
24. l\_objGridDropbox.DisplayMember = "PaidWith";
25. }
27. }
29. }

Find Latitude and Longitude of a System

<http://www.c-sharpcorner.com/blogs/find-latitude-and-longitude-of-a-system>

1. This only works in .NET Framework 4.0 or later and Windows 7 or later.
2. The Location API is defined in the System.Device DLL so add a reference to that library. The code includes the following using directive to make using the Location API easier.

Get All Sheets From Excel And Fill Data Of Selected Excel Sheet In DataGridView

<http://www.c-sharpcorner.com/blogs/get-all-sheets-from-excel-and-fill-data-of-selected-excel-sheet-in-datagridview>

               DataTable dt = OleDbcon.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, null);

1. OleDbDataAdapter oledbDa = **new** OleDbDataAdapter("Select \* from [" + comboBox1.Text + "$]", OleDbcon);
2. DataTable dt = **new** DataTable();
4. oledbDa.Fill(dt);
6. dataGridView1.DataSource = dt;

## [线程之间的通讯---SynchronizationContext](http://www.cnblogs.com/Kevin-moon/archive/2009/01/13/1374353.html)

<http://www.cnblogs.com/Kevin-moon/archive/2009/01/13/1374353.html>

SynchronizationContext.Current 只存在与UI线程，其他线程调用后得到null

<http://www.cnblogs.com/Kevin-moon/archive/2009/01/16/1376812.html>

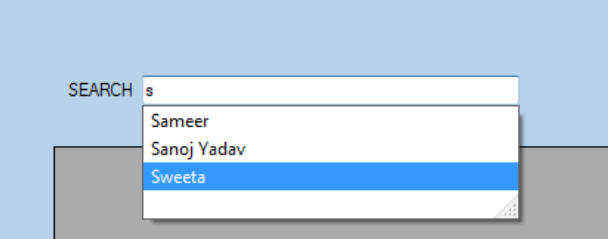
Making A Simple Non-Freezing Window Forms Application

<http://www.c-sharpcorner.com/UploadFile/27c648/making-a-simple-non-freezing-window-forms-application/>

1. **private** async **void** button2\_Click(**object** sender, EventArgs e)
2. {
3. button1.Enabled = **false**;
4. button2.Enabled = **false**;
5. var count = 0;
7. await Task.Run(() =>
8. {
9. **for** (var i = 0; i <= 1000000; i++)
10. {
11. UpdateUI(i);
12. count = i;
13. }
14. });
15. label1.Text = @"Count : " + count;
16. button1.Enabled = **true**;
17. button1.Enabled = **false**;
18. }

AutoComplete TextBox In Windows Form

<http://www.c-sharpcorner.com/UploadFile/33b051/autocomplete-textbox-in-windows-form/>



AutoCompleteStringCollection coll = new AutoCompleteStringCollection();

textBox1.AutoCompleteMode = AutoCompleteMode.Suggest;

textBox1.AutoCompleteSource = AutoCompleteSource.CustomSource;

textBox1.AutoCompleteCustomSource = coll;

# AutoComplete ComboBox in DataGridView using C#.net Windows Application

http://www.c-sharpcorner.com/uploadfile/aa04e6/autocomplete-combobox-in-datagridview-using-C-Sharp-net-windows-application/

DataGridViewComboBoxColumn combo2 = new DataGridViewComboBoxColumn();  
            combo2.HeaderText = "Types of Jobs";  
            combo2.Items.Add("Accounting");  
            combo2.Items.Add("HR");  
            combo2.Items.Add("Finance");  
            combo2.Items.Add("Transportation");  
            combo2.Items.Add("Testing");  
            dataGridView1.Columns.Add(combo2);

Now Click on DataGridView and generate EditingControlShowing event and write the folllowing code in it:

private void dataGridView1\_EditingControlShowing(object sender, DataGridViewEditingControlShowingEventArgs e)

{

if (e.Control is DataGridViewComboBoxEditingControl)

{

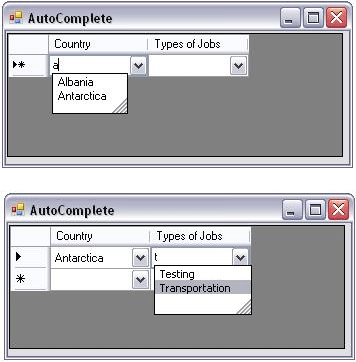
((ComboBox)e.Control).DropDownStyle = ComboBoxStyle.DropDown;

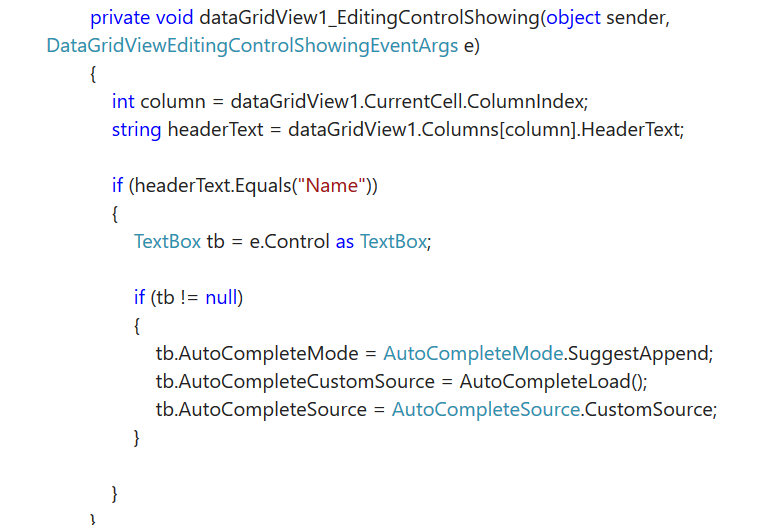
((ComboBox)e.Control).AutoCompleteSource = AutoCompleteSource.ListItems;

((ComboBox)e.Control).AutoCompleteMode = System.Windows.Forms.AutoCompleteMode.Suggest;

}

}





# DataGridView.EditingControlShowing 事件

可以处理此事件，以便在单元格进入编辑模式时对编辑控件执行自定义初始化。要自定义控件的显示特征，请设置 [DataGridViewEditingControlShowingEventArgs.CellStyle](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.datagridvieweditingcontrolshowingeventargs.cellstyle%28v=vs.80%29.aspx) 属性返回的对象的属性，而不要设置 [DataGridViewEditingControlShowingEventArgs.Control](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.datagridvieweditingcontrolshowingeventargs.control%28v=vs.80%29.aspx) 属性返回的控件的属性。

private void dataGridView1\_EditingControlShowing(object sender,

DataGridViewEditingControlShowingEventArgs e)

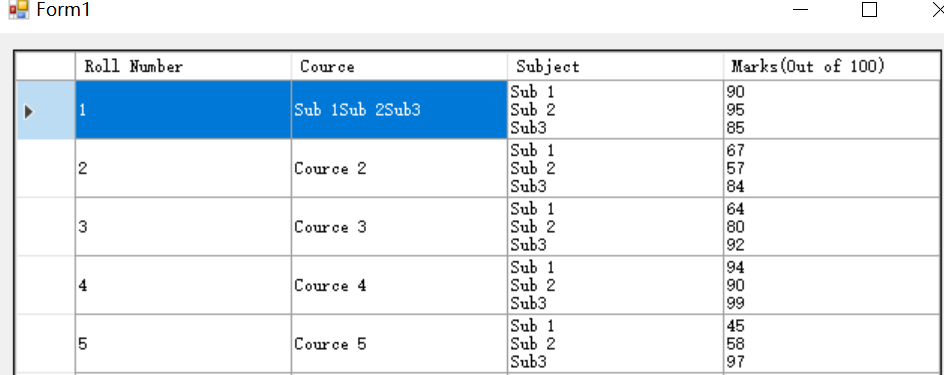
{

e.CellStyle.BackColor = Color.Aquamarine;

}

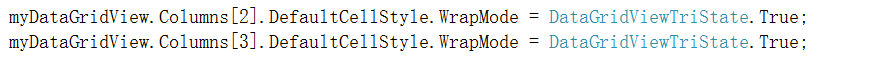
# Hierarchical Data Into DataGridView in C#

<http://www.c-sharpcorner.com/UploadFile/9f4ff8/hierarchical-data-into-datagridview-in-C-Sharp/>



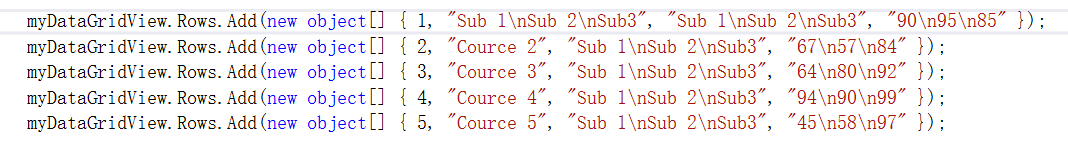
For that you have to use the WrapMode property. Using the WrapMode Property we can enter the data into the new line in the particular cell.

For that you have to set that WrapMode property as True like below:



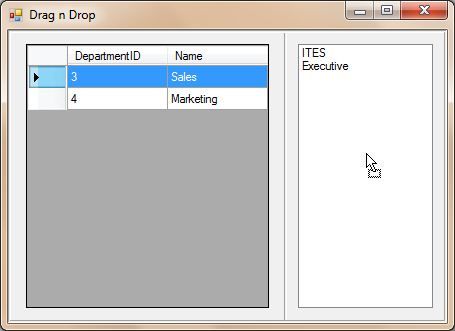
In our case we have last the columns (column 2 & column 3) in which we will use that WrapMode property as true.

Now start entering the data into the DataGridView one by one.



# Drag and Drop From DataGridView to ListBox in a Windows Forms Application

<http://www.c-sharpcorner.com/UploadFile/deepak.sharma00/drag-and-drop-from-datagridview-to-listbox-in-a-windows-form/>



Set the AllowDrop property of the ListBox to true, as in:  
listBox1.AllowDrop = true;

Write the following in the MouseDown event of the DataGridView:

private void dataGridView1\_MouseDown(object sender, MouseEventArgs e)

{

dataGridView1.DoDragDrop(dataGridView1.SelectedRows, DragDropEffects.Move);

}

The first parameter is the data that we want to send in drag and drop operation

Write the following in the DragEnter event of the ListBox:

private void listBox1\_DragEnter(object sender, DragEventArgs e)

{

if (e.Data.GetDataPresent(typeof(DataGridViewSelectedRowCollection)))

{

e.Effect = DragDropEffects.Move;

}

}

DoDragDrop（）和dragEnter事件里的effect 要一样？

Write the following in the DragDrop event of the ListBox:

private void listBox1\_DragDrop(object sender, DragEventArgs e)

{

DataGridViewSelectedRowCollection rows = (DataGridViewSelectedRowCollection)e.Data.GetData(typeof(DataGridViewSelectedRowCollection));

foreach (DataGridViewRow row in rows)

{

listBox1.Items.Add(row.Cells[1].Value);

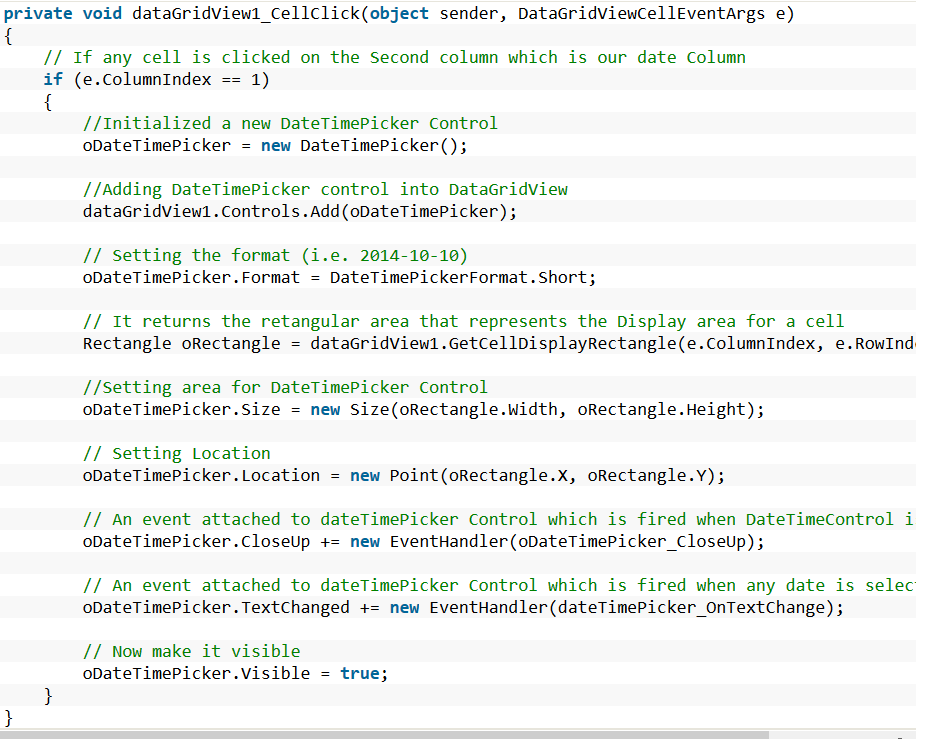
dataGridView1.Rows.Remove(row);

}

}

# Embedding Calendar (DateTimePicker) Control Into DataGridView Cell

<http://www.c-sharpcorner.com/UploadFile/0f68f2/embedding-calendar-datetimepicker-control-into-datagridvie586/>



// 摘要:

// 在显示用于编辑单元格的控件时发生。

public event DataGridViewEditingControlShowingEventHandler EditingControlShowing;

// grid Editing Control Showing

private void dShanuDGV\_EditingControlShowing(object sender, DataGridViewEditingControlShowingEventArgs e)

{

if (listcolumnIndex.Contains(shanuDGVs.CurrentCell.ColumnIndex))

{

TextBox itemID = e.Control as TextBox;

if (itemID != null)

{

itemID.KeyPress += new KeyPressEventHandler(itemID\_KeyPress);

}

}

}

*DataSet dataSet = new DataSet();*

*dataSet.ReadXml(@"C:\Books\Books.xml");*

*dataGridView1.DataSource = dataSet.Tables[0];*

DataGridViewComboBoxCell EmployeeIDCell = new DataGridViewComboBoxCell();

/\* In this case the employeeID is fixed from 1 to 9. Use Items.Add method to add,elements to the combo box cell \*/  
for(int i = 1;i<=9;i++)

EmployeeIDCell.Items.Add(i);

/\* Create a new Combo Box Column and set its CellTemplate property to, EmployeeIDCell \*/   
DataGridViewComboBoxColumn EmpIdColumn = new DataGridViewComboBoxColumn();  
EmpIdColumn.CellTemplate [[1]](#footnote-1)[1] = EmployeeIDCell;

// Create a new Combo Box Column

DataGridViewComboBoxColumn EmpIdColumn = new DataGridViewComboBoxColumn();  
// Set the DataSource of EmpIdColumn as bellow

EmpIdColumn.DataSource = myDataSet.Tables[0];

// Set the ValueMember property as done bellow

EmpIdColumn.ValueMember = myDataSet.Tables[0].Columns[0].ColumnName.ToString();

# Filter data dispalyed in a DataGridView using BindingSource and DataView

http://www.c-sharpcorner.com/uploadfile/yougerthen/filter-data-dispalyed-in-a-datagridview-using-bindingsource-and-dataview/

myBindingSource = new BindingSource();

myBindingSource.DataSource = myDataSet;

myBindingSource.DataMember = myDataSet.Tables[0].TableName;

myGridView.DataSource = myBindingSource;

if (PriceCombo.Text == "0") myBindingSource.Filter = "UnitPrice <= 0";

if (PriceCombo.Text == "10") myBindingSource.Filter = "UnitPrice <= 10";

myDataView = new DataView();

myDataView.Table = myDataSet.Tables[0];

myGridView.DataSource = myDataView;

if (PriceCombo.Text == "0") myDataView.RowFilter = "UnitPrice <= 0";

if (PriceCombo.Text == "10") myDataView.RowFilter = "UnitPrice <= 10";

BindingList<T> Class

The BindingList<T> class can be used as a base class to create a two-way data-binding mechanism. BindingList<T> provides a concrete, generic implementation of the [IBindingList](https://msdn.microsoft.com/en-us/library/system.componentmodel.ibindinglist(v=vs.110).aspx) interface. This is an alternative to implementing the complete [IBindingList](https://msdn.microsoft.com/en-us/library/system.componentmodel.ibindinglist(v=vs.110).aspx) interface, which can be difficult because of the subtle interaction between [IBindingList](https://msdn.microsoft.com/en-us/library/system.componentmodel.ibindinglist(v=vs.110).aspx), [IEditableObject](https://msdn.microsoft.com/en-us/library/system.componentmodel.ieditableobject(v=vs.110).aspx), and the associated [CurrencyManager](https://msdn.microsoft.com/en-us/library/system.windows.forms.currencymanager(v=vs.110).aspx). However, the typical solutions programmer will use a class that provides data binding functionality, such as [BindingSource](https://msdn.microsoft.com/en-us/library/system.windows.forms.bindingsource(v=vs.110).aspx), instead of directly using BindingList<T>.

BindingList<T> supports factory-created instances through the extensible [AddNew](https://msdn.microsoft.com/en-us/library/ms132687(v=vs.110).aspx) method. (This same type of extensibility is also found in other classes, such as [BindingSource](https://msdn.microsoft.com/en-us/library/system.windows.forms.bindingsource(v=vs.110).aspx)) In addition, since this class implements the [ICancelAddNew](https://msdn.microsoft.com/en-us/library/system.componentmodel.icanceladdnew(v=vs.110).aspx) interface, it enables transactional commits or rollbacks of the new item through the [EndNew](https://msdn.microsoft.com/en-us/library/ms132693(v=vs.110).aspx) and [CancelNew](https://msdn.microsoft.com/en-us/library/ms132691(v=vs.110).aspx) methods.

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

{

Part newPart = listOfParts.AddNew();

if (newPart.PartName.Contains(" "))

{

MessageBox.Show("Part names cannot contain spaces.");

listOfParts.CancelNew(listOfParts.IndexOf(newPart));

}

else

{

textBox2.Text = randomNumber.Next(9999).ToString();

textBox1.Text = "Enter part name";

}

}

// Create a new part from the text in the two text boxes.

void listOfParts\_AddingNew(object sender, AddingNewEventArgs e)

{

e.NewObject = new Part(textBox1.Text, int.Parse(textBox2.Text));

}

void listOfParts\_ListChanged(object sender, ListChangedEventArgs e)

{

MessageBox.Show(e.ListChangedType.ToString());

}

DataGridViewColumn.CellTemplate 属性

private void CustomizeCellsInThirdColumn()

{

int thirdColumn = 2;

DataGridViewColumn column =

dataGridView.Columns[thirdColumn];

DataGridViewCell cell = new DataGridViewTextBoxCell();

cell.Style.BackColor = Color.Wheat;

column.CellTemplate = cell;

}

Changing the properties of the cell template will not immediately affect the user interface (UI) of the column's existing cells

# Custom Time Cell in DataGridView

<http://www.c-sharpcorner.com/uploadfile/ankurmee/custom-time-cell-in-datagridview/>

新增继承自DataGridViewTextBoxCell 的自定义单元格CalendarCell1

新增自定义单元格的编辑控件CalendarEditingControl1，继承自IDataGridViewEditingControl

新增继承自DataGridViewColumn 的自定义列。列的CellTemplate赋值为CalendarCell1

# DataGridView.EditingControl 属性

获取当前单元格承载的控件（如果包含编辑控件的单元格处于编辑模式下）。

下面的代码示例阐释如何在自定义单元格类型的重写方法中使用此属性。在该示例中，检索对编辑控件的引用，将该控件强制转换为自定义编辑控件类型，然后在编辑控件中填入单元格的当前值。复制

public override void InitializeEditingControl(int rowIndex, object

initialFormattedValue, DataGridViewCellStyle dataGridViewCellStyle)

{

// Set the value of the editing control to the current cell value.

base.InitializeEditingControl(rowIndex, initialFormattedValue,

dataGridViewCellStyle);

CalendarEditingControl ctl =

DataGridView.EditingControl as CalendarEditingControl;

ctl.Value = (DateTime)this.Value;

}

DataGridViewCell.InitializeEditingControl 方法 (Int32, Object, DataGridViewCellStyle)

initialFormattedValue

Type: [System.Object](https://msdn.microsoft.com/zh-cn/library/system.object(v=vs.110).aspx)

An [Object](https://msdn.microsoft.com/zh-cn/library/system.object(v=vs.110).aspx) that represents the value displayed by the cell when editing is started.

As an optimization technique, typically all the cells of the same type and in the same [DataGridView](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.datagridview(v=vs.110).aspx) share a single hosted editing control. However, before the control is used by a cell, it needs to be initialized by the InitializeEditingControl method. The first time it is called, this method adds the control to the list of editing controls in its parent [DataGridView](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.datagridview(v=vs.110).aspx). It also initializes some of the visual properties of the cell. For example, InitializeEditingControl sets the background color of the editing area to match the supplied cell style parameter. Subsequent calls to InitializeEditingControl do nothing.

public class CalendarCell : DataGridViewTextBoxCell

{

public CalendarCell()

: base()

{

// Use the short date format.

this.Style.Format = "d";

}

public override void InitializeEditingControl(int rowIndex, object

initialFormattedValue, DataGridViewCellStyle dataGridViewCellStyle)

{

// Set the value of the editing control to the current cell value.

base.InitializeEditingControl(rowIndex, initialFormattedValue,

dataGridViewCellStyle);

CalendarEditingControl ctl =

DataGridView.EditingControl as CalendarEditingControl;

// Use the default row value when Value property is null.

if (this.Value == null)

{

ctl.Value = (DateTime)this.DefaultNewRowValue;

}

else

{

ctl.Value = (DateTime)this.Value;

}

}

public override Type EditType

{

get

{

// Return the type of the editing control that CalendarCell uses.

return typeof(CalendarEditingControl);

}

}

public override Type ValueType

{

get

{

// Return the type of the value that CalendarCell contains.

return typeof(DateTime);

}

}

public override object DefaultNewRowValue

{

get

{

// Use the current date and time as the default value.

return DateTime.Now;

}

}

}

dataGridView1.SelectionMode = DataGridViewSelectionMode.CellSelect;

dataGridView1.MultiSelect = true;

Dynamically select the rows through code behind itself.

dataGridView1[1,1].Selected = true;

dataGridView1[2, 1].Selected = true;

# How to : Add an autonumber column in a DataGridView

<http://www.c-sharpcorner.com/uploadfile/santowebster/how-to-add-an-autonumber-column-in-a-datagridview/>

private DataTable AutoNumberedTable(DataTable SourceTable)

{

DataTable ResultTable = new DataTable();

DataColumn AutoNumberColumn = new DataColumn();

AutoNumberColumn.ColumnName="S.No.";

AutoNumberColumn.DataType = typeof(int);

AutoNumberColumn.AutoIncrement = true;

AutoNumberColumn.AutoIncrementSeed = 1;

AutoNumberColumn.AutoIncrementStep = 1;

ResultTable.Columns.Add(AutoNumberColumn);

ResultTable.Merge(SourceTable);

return ResultTable;

}



# SqlConnectionStringBuilder Class

<https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnectionstringbuilder(VS.80).aspx>

SqlConnectionStringBuilder builder =

new SqlConnectionStringBuilder(GetConnectionString());

// Pass the SqlConnectionStringBuilder an existing

// connection string, and you can retrieve and

// modify any of the elements.

builder.ConnectionString = "server=(local);user id=ab;" +

"password= a!Pass113;initial catalog=AdventureWorks";

// Now that the connection string has been parsed,

// you can work with individual items.

Console.WriteLine(builder.Password);

builder.Password = "new@1Password";

builder.AsynchronousProcessing = true;

// You can refer to connection keys using strings,

// as well. When you use this technique (the default

// Item property in Visual Basic, or the indexer in C#),

// you can specify any synonym for the connection string key

// name.

builder["Server"] = ".";

builder["Connect Timeout"] = 1000;

builder["Trusted\_Connection"] = true;

Console.WriteLine(builder.ConnectionString);

# DataGridView.OnCellValidating 方法 (DataGridViewCellValidatingEventArgs)

<https://msdn.microsoft.com/zh-cn/library/system.windows.forms.datagridview.oncellvalidating(v=vs.120).aspx>

引发 [CellValidating](https://msdn.microsoft.com/zh-cn/library/system.windows.forms.datagridview.cellvalidating%28v=vs.110%29.aspx) 事件。

引发事件时，将通过委托调用事件处理程序。 有关详细信息，请参阅 [NIB ︰ 引发事件](https://msdn.microsoft.com/zh-cn/library/wkzf914z%28v=vs.110%29.aspx)。

OnCellValidating 方法还允许派生类对事件进行处理而不必附加委托。 这是在派生类中处理事件的首选技术。

当重写 OnCellValidating 在派生类中，一定要调用基类的 OnCellValidating 方法，以便已注册的委托对事件进行接收。

1. [↑](#footnote-ref-1)