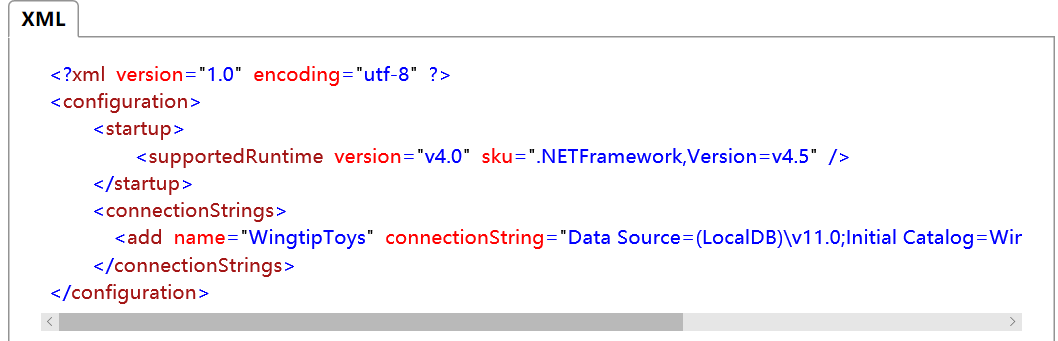
The T:System.Configuration.ConfigurationManager class enables you to access machine, application, and user configuration information.

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>

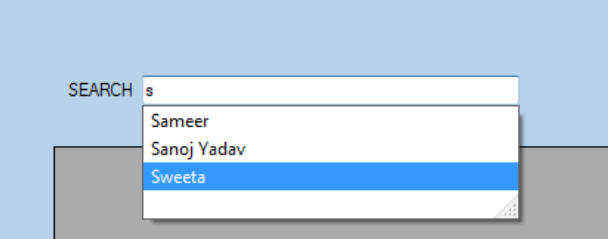
Making A Simple Non-Freezing Window Forms Application

<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;

# 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;

}