[**C# ListView用法详解**](https://www.cnblogs.com/shadowme/p/6250070.html)

**https://www.cnblogs.com/shadowme/p/6250070.html**

[**C# ListView用法详解**](https://www.cnblogs.com/shadowme/p/6250070.html)

**一、ListView类**

**1、常用的基本属性：**

（1）[FullRowSelect](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_FullRowSelect.htm)：设置是否行选择模式。(默认为false) 提示：只有在Details视图该属性才有意义。

（2） [GridLines](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_GridLines.htm)：设置行和列之间是否显示网格线。（默认为false）提示：只有在Details视图该属性才有意义。

（3）[AllowColumnReorder](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_AllowColumnReorder.htm)：设置是否可拖动列标头来对改变列的顺序。（默认为false）提示：只有在Details视图该属性才有意义。

（4）[View](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_View.htm)：获取或设置项在控件中的显示方式，包括Details、LargeIcon、List、SmallIcon、Tile（默认为 LargeIcon）

（5）[MultiSelect](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_MultiSelect.htm)：设置是否可以选择多个项。（默认为false）

（6）[HeaderStyle](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_HeaderStyle.htm)：获取或设置列标头样式。

Clickable：列标头的作用类似于按钮，单击时可以执行操作（例如排序）。

NonClickable：列标头不响应鼠标单击。

None：不显示列标头。

（7）[LabelEdit](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_LabelEdit.htm)：设置用户是否可以编辑控件中项的标签，对于Detail视图，只能编辑行第一列的内容。（默认为false）

（8）[CheckBoxes](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_CheckBoxes.htm)：设置控件中各项的旁边是否显示复选框。（默认为false）

（9）[LargeImageList](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_LargeImageList.htm)：大图标集。提示：只在LargeIcon视图使用。

（10）[SmallImageList](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_SmallImageList.htm)：小图标集。提示：只有在SmallIcon视图使用。

（11）[StateImageList](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_StateImageList.htm)：图像蒙板。这些图像蒙板可用作LargeImageList和SmallImageList图像的覆盖图，这些图像可用于指示项的应用程序定义的状态。（暂时不大懂）

（12）[SelectedItems](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_SelectedItems.htm)：获取在控件中选定的项。

（13）[CheckedItems](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_CheckedItems.htm)：获取控件中当前复选框选中的项。

（14）[Sorit](http://blog.csdn.net/czw2010/article/details/M_System_Windows_Forms_ListView_Sort.htm)[ng](http://write.blog.csdn.net/postedit/M_System_Windows_Forms_ListView_Sort.htm)：对列表视图的项进行排序。(默认为None)

Ascending：项按递增顺序排序。

Descending：项按递减顺序排序。

None：项未排序。

（15）[Scrollable](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_Scrollable.htm)：设置当没有足够空间来显示所有项时是否显示滚动条。（默认为true)

（16）[HoverSelection](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_HoverSelection.htm)：设置当鼠标指针悬停于项上时是否自动选择项。（默认为false）

（17）[HotTracking](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_HotTracking.htm)：设置当鼠标指针经过项文本时，其外观是否变为超链接的形式。（默认为false）

（18）[HideSelection](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_HideSelection.htm)：设置选定项在控件没焦点时是否仍突出显示。（默认为false）

（19）[ShowGroups](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_ShowGroups.htm)：设置是否以分组方式显示项。（默认为false);

（20）[Groups](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_Groups.htm)：设置分组的对象集合。

（21）[TopItem](http://blog.csdn.net/czw2010/article/details/P_System_Windows_Forms_ListView_TopItem.htm)：获取或设置控件中的第一个可见项，可用于定位。（效果类似于[EnsureVisible](http://write.blog.csdn.net/postedit/M_System_Windows_Forms_ListView_EnsureVisible_1_804c5d7d.htm)方法）

**2、常用方法：**

（1）[BeginUpdate](http://blog.csdn.net/czw2010/article/details/M_System_Windows_Forms_ListView_BeginUpdate.htm)：避免在调用[EndUpdate](http://blog.csdn.net/czw2010/article/details/M_System_Windows_Forms_ListView_EndUpdate.htm) 方法之前描述控件。当插入大量数据时，可以有效地避免控件闪烁，并能大大提高速度。

（2）[EndUpdate](http://blog.csdn.net/czw2010/article/details/M_System_Windows_Forms_ListView_EndUpdate.htm)：在[BeginUpdate](http://blog.csdn.net/czw2010/article/details/M_System_Windows_Forms_ListView_BeginUpdate.htm) 方法挂起描述后，继续描述列表视图控件。（结束更新）

（3）[EnsureVisible](http://blog.csdn.net/czw2010/article/details/M_System_Windows_Forms_ListView_EnsureVisible_1_804c5d7d.htm)：列表视图滚动定位到指定索引项的选项行。（效果类似于[TopItem](http://write.blog.csdn.net/postedit/P_System_Windows_Forms_ListView_TopItem.htm)属性）

（4）[FindItemWithText](http://blog.csdn.net/czw2010/article/details/O_T_System_Windows_Forms_ListView_FindItemWithText.htm)：查找以给定文本值开头的第一个 ListViewItem。

（5）[FindNearestItem](http://blog.csdn.net/czw2010/article/details/O_T_System_Windows_Forms_ListView_FindNearestItem.htm)：按照指定的搜索方向，从给定点开始查找下一个项。提示：只有在LargeIcon或SmallIcon视图才能使用该方法。

**3、常用事件：**

（1）[AfterLabelEdit](http://blog.csdn.net/czw2010/article/details/E_System_Windows_Forms_ListView_AfterLabelEdit.htm)：当用户编辑完项的标签时发生，需要[LabelEdit](http://write.blog.csdn.net/postedit/P_System_Windows_Forms_ListView_LabelEdit.htm)属性为true。

（2）[BeforeLabelEdit](http://blog.csdn.net/czw2010/article/details/E_System_Windows_Forms_ListView_BeforeLabelEdit.htm)：当用户开始编辑项的标签时发生。

（3）[ColumnClick](http://blog.csdn.net/czw2010/article/details/E_System_Windows_Forms_ListView_ColumnClick.htm)：当用户在列表视图控件中单击列标头时发生。

**二、ListView的五种视图：**

1、**LargeIcon**：每个项都显示为一个最大化图标，在它的下面有一个标签。（效果见下图）

2、**SmallIcon**：每个项都显示为一个小图标，在它的右边带一个标签。（效果见下图）

3、**List**：每个项都显示为一个小图标，在它的右边带一个标签。各项排列在列中，没有列标头。（效果见下图）

4、**Details**：可以显示任意的列，但只有第一列可以包含一个小图标和标签，其它的列项只能显示文字信息，有列表头。（效果见下图）

5、**Tile**：每个项都显示为一个完整大小的图标，在它的右边带项标签和子项信息。（只有Windows XP 和 Windows Server 2003 系列支持）

**①Details视图：**

this.listView1.SmallImageList = this.imageList1; //将listView的图标集与imageList1绑定

**（1）列表头创建（记得，需要先创建列表头）**

[复制代码](javascript:void(0);)

ColumnHeader ch = new ColumnHeader();

ch.Text = "列标题1"; //设置列标题

ch.Width = 120; //设置列宽度

ch.TextAlign = HorizontalAlignment.Left; //设置列的对齐方式

this.listView1.Columns.Add(ch); //将列头添加到ListView控件。

[复制代码](javascript:void(0);)

**或者**

this.listView1.Columns.Add("列标题1", 120, HorizontalAlignment.Left); //一步添加

**（2）添加数据项**

[复制代码](javascript:void(0);)

this.listView1.BeginUpdate(); //数据更新，UI暂时挂起，直到EndUpdate绘制控件，可以有效避免闪烁并大大提高加载速度

for (int i = 0; i < 10; i++) //添加10行数据

{

ListViewItem lvi = new ListViewItem();

lvi.ImageIndex = i; //通过与imageList绑定，显示imageList中第i项图标

lvi.Text = "subitem" + i;

lvi.SubItems.Add("第2列,第"+i+"行");

lvi.SubItems.Add("第3列,第"+i+"行");

this.listView1.Items.Add(lvi);

}

this.listView1.EndUpdate(); //结束数据处理，UI界面一次性绘制。

[复制代码](javascript:void(0);)

**（3）显示项**

[复制代码](javascript:void(0);)

foreach (ListViewItem item in this.listView1.Items)

{

//处理行

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

{

//处理列

MessageBox.Show(item.SubItems[i].Text);

}

}

[复制代码](javascript:void(0);)

**（4）移除某项**

foreach (ListViewItem lvi in listView1.SelectedItems) //选中项遍历

{

listView1.Items.RemoveAt(lvi.Index); // 按索引移除

//listView1.Items.Remove(lvi); //按项移除

}

**（5）行高设置（利用imageList实现）**

ImageList imgList = new ImageList();

imgList.ImageSize = new Size(1, 20);// 设置行高 20 //分别是宽和高

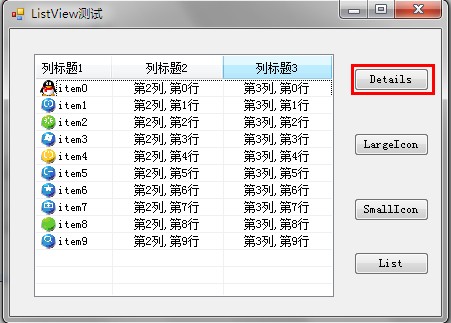
listView1.SmallImageList = imgList; //这里设置listView的SmallImageList ,用imgList将其撑大

**（6）清空**

this.listView1.Clear(); //从控件中移除所有项和列（包括列表头）。

this.listView1.Items.Clear(); //只移除所有的项。

运行效果：



**②largeIcon视图：**

[复制代码](javascript:void(0);)

this.listView1.View = View.LargeIcon;

this.listView1.LargeImageList = this.imageList2;

this.listView1.BeginUpdate();

for (int i = 0; i < 10; i++)

{

ListViewItem lvi = new ListViewItem();

lvi.ImageIndex = i;

lvi.Text = "item" + i;

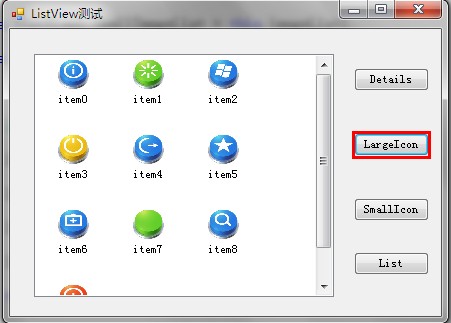
this.listView1.Items.Add(lvi);

}

this.listView1.EndUpdate();

[复制代码](javascript:void(0);)

运行效果：



**③SmallIcon视图：**

[复制代码](javascript:void(0);)

this.listView1.View = View.SmallIcon;

this.listView1.SmallImageList= this.imageList1;

this.listView1.BeginUpdate();

for (int i = 0; i < 10; i++)

{

ListViewItem lvi = new ListViewItem();

lvi.ImageIndex = i;

lvi.Text = "item" + i;

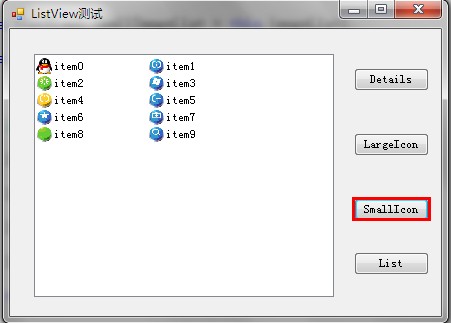
this.listView1.Items.Add(lvi);

}

this.listView1.EndUpdate();

[复制代码](javascript:void(0);)

运行效果：



**④List视图：**

[复制代码](javascript:void(0);)

this.listView1.View = View.List;

this.listView1.SmallImageList= this.imageList1;

this.listView1.BeginUpdate();

for (int i = 0; i < 10; i++)

{

ListViewItem lvi = new ListViewItem();

lvi.ImageIndex = i;

lvi.Text = "item" + i;

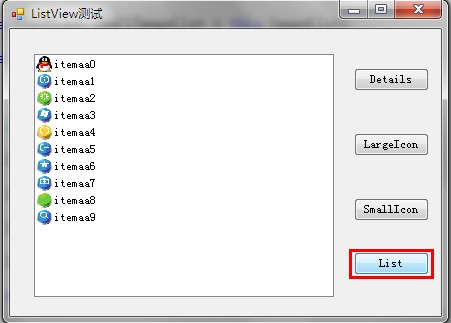
this.listView1.Items.Add(lvi);

}

this.listView1.EndUpdate();

[复制代码](javascript:void(0);)

运行效果：



**三、其它应用**

**1、分组：**

[复制代码](javascript:void(0);)

ListViewGroup man\_lvg = new ListViewGroup(); //创建男生分组

man\_lvg.Header = "男生"; //设置组的标题。

//man\_lvg.Name = "man"; //设置组的名称。

man\_lvg.HeaderAlignment = HorizontalAlignment.Left; //设置组标题文本的对齐方式。（默认为Left）

ListViewGroup women\_lvg = new ListViewGroup(); //创建女生分组

women\_lvg.Header = "女生";

//women\_lvg.Name = "women";

women\_lvg.HeaderAlignment = HorizontalAlignment.Center; //组标题居中对齐

this.listView1.Groups.Add(man\_lvg); //把男生分组添加到listview中

this.listView1.Groups.Add(women\_lvg); //把男生分组添加到listview中

this.listView1.ShowGroups = true; //记得要设置ShowGroups属性为true（默认是false），否则显示不出分组

for (int i = 0; i < 5; i++)

{

ListViewItem lvi = new ListViewItem();

lvi.ImageIndex = i;

lvi.Text = "item"+i;

lvi.ForeColor = Color.Blue; //设置行颜色

lvi.SubItems.Add("第2列,第"+i+"行");

lvi.SubItems.Add("第3列,第"+i+"行");

man\_lvg.Items.Add(lvi); //分组添加子项

// 或 lvi.Group = man\_lvg; //分组添加子项

this.listView1.Items.Add(lvi);

}

[复制代码](javascript:void(0);)

运行效果:



**2、查找文本（只能查找到匹配前缀的文本且只能找出第一个符合的项）：**

[复制代码](javascript:void(0);)

ListViewItem foundItem= this.listView1.FindItemWithText(this.textBox1.Text,true,0); //参数1：要查找的文本；参数2：是否子项也要查找；参数3：开始查找位置

if (foundItem != null)

{

this.listView1.TopItem = foundItem; //定位到该项

foundItem.ForeColor = Color.Red;

}