

C# 원폼 프로그래밍, ListView Control

ListView 컨트롤은 항목의 컬렉션을 표시하는 데 사용되는데 ListBox 컨트롤보다 정교한 컨트롤이다. 여러 여러 칼럼을 가진 데이터 보기에 유리하다. 리스트 형태를 지정하는 View 속성은 Details, List, Tile, LargeIcon, SmallIcon 이 있다.

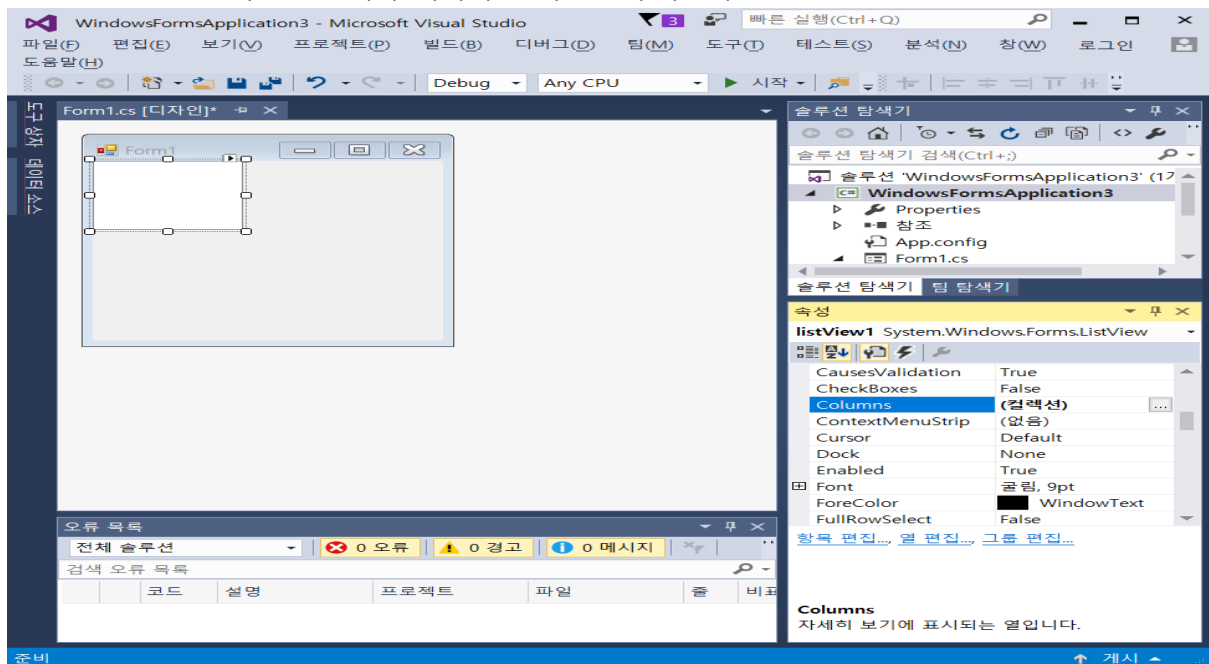
[ListView에 칼럼 추가]

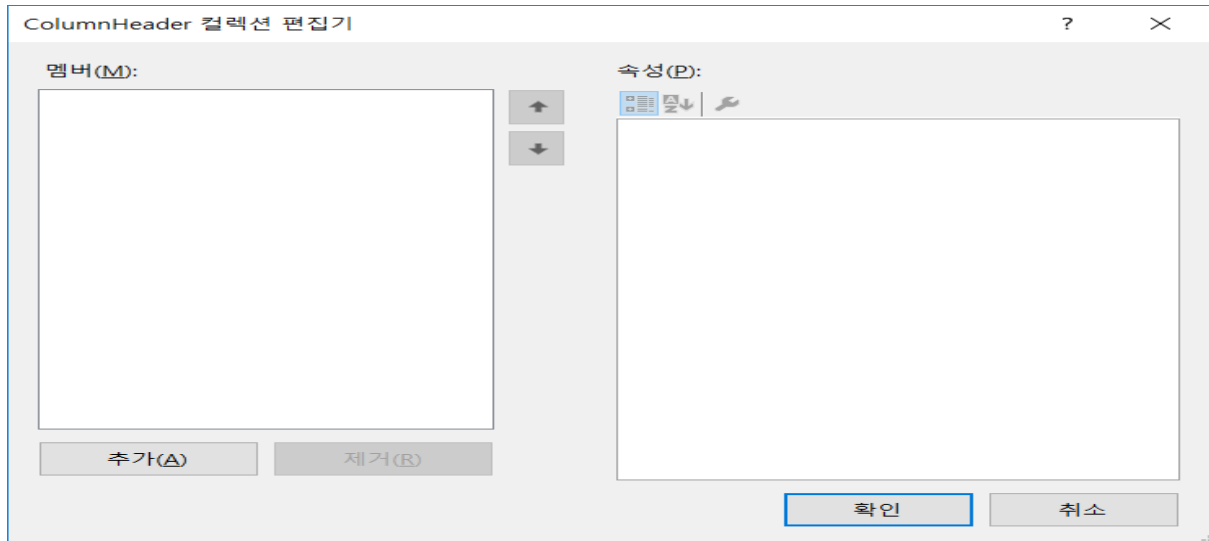
- 원폼 디자이너에서 추가

디자이너에서 열을 추가하려면 컨트롤의 View 속성을 Details로 설정.

속성 창에서 Columns 속성 옆에 있는 줄임표 단추(...)를 클릭.

ColumnHeader 컬렉션 편집기가 나타나면 새열을 추가한다.





- 소스코드에서 추가

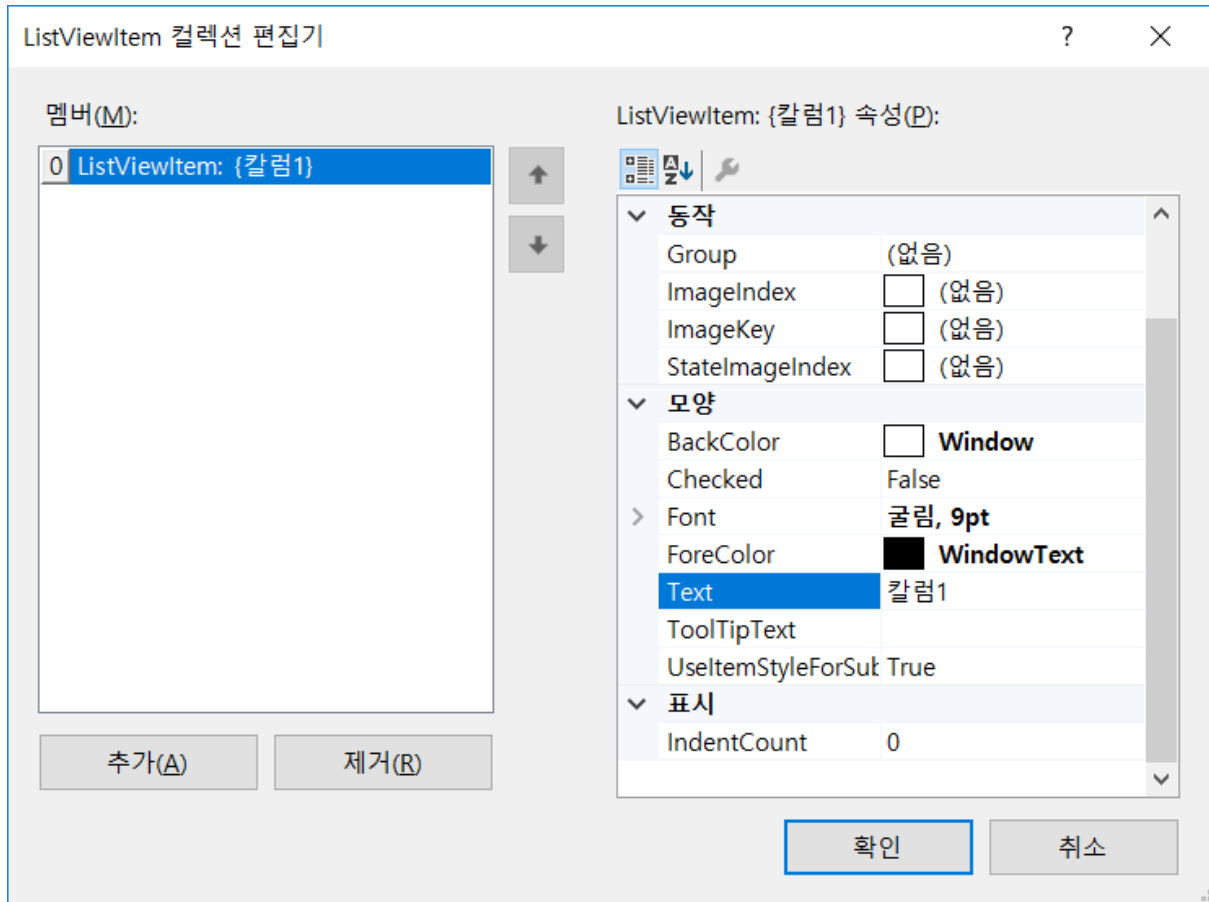
컨트롤의 View 속성을 Details로 설정하고 Columns 속성의 Add 메서드를 사용한다.

```
listView1.View = View.Details;
// Add a column with width 20 and left alignment.
listView1.Columns.Add("컬럼1", 20, HorizontalAlignment.Left);
```

[ListView에 항목 추가]

- 윈폼 디자이너에서 추가

속성 창에서 Items 속성 옆에 있는 줄임표 단추(...)를 클릭 -> ListViewItem 컬렉션 편집기가 나타나면 추가 버튼을 클릭한다. ext 및 ImageIndex 속성과 같은 새 항목의 속성을 설정할 수 있다. 항목을 제거하려면 항목을 선택한 다음 제거 단추를 클릭하면 된다.



- 소스코드에서 추가, 삭제

```
// Adds a new item with ImageIndex 3
listView1.Items.Add("List item text", 3);
```

Items 속성의 RemoveAt 메서드 또는 Clear 메서드를 사용한다. RemoveAt 메서드는 항목 하나를 제거하며 Clear 메서드는 목록에 있는 항목을 모두 제거한다.

```
// Removes the first item in the list.
listView1.Items.RemoveAt(0);
// Clears all the items.
listView1.Items.Clear();
```

[예제]


```
actresses.Add(new Actress("Jessica Alba", 1981));
actresses.Add(new Actress("Angelina Jolie", 1975));
actresses.Add(new Actress("Natalie Portman", 1981));
actresses.Add(new Actress("Rachel Weiss", 1971));
actresses.Add(new Actress("Scarlett Johansson", 1984));
```

```
ColumnHeader name = new ColumnHeader();
name.Text = "Name";
name.Width = -1;
ColumnHeader year = new ColumnHeader();
year.Text = "Year";
```

```
SuspendLayout();
```

```
ListView lv = new ListView();
lv.Parent = this;
lv.FullRowSelect = true;
lv.GridLines = true;
lv.AllowColumnReorder = true;
lv.Sorting = SortOrder.Ascending;
lv.Columns.AddRange(new ColumnHeader[] { name, year });
lv.ColumnClick += new ColumnClickEventHandler(ColumnClick);
```

```
foreach (Actress act in actresses)
{
    ListViewItem item = new ListViewItem();
    item.Text = act.name;
    item.SubItems.Add(act.year.ToString());
    lv.Items.Add(item);
}
```

```
lv.Dock = DockStyle.Fill;
lv.Click += new EventHandler(OnChanged);
```

```
sb = new StatusBar();
sb.Parent = this;
lv.View = View.Details;
```

```
ResumeLayout();
```

```
CenterToScreen();
```

```
}
```

```
void OnChanged(object sender, EventArgs e)
{
    ListView lv = (ListView)sender;
    string name = lv.SelectedItems[0].SubItems[0].Text;
    string born = lv.SelectedItems[0].SubItems[1].Text;
    sb.Text = name + ", " + born;
}
```

```
void ColumnClick(object sender, ColumnClickEventArgs e)
{
    ListView lv = (ListView)sender;

    if (lv.Sorting == SortOrder.Ascending)
    {
        lv.Sorting = SortOrder.Descending;
    }
    else
    {
        lv.Sorting = SortOrder.Ascending;
    }
}

}

class MApplication
{
    public static void Main()
    {
        Application.Run(new MForm());
    }
}
```