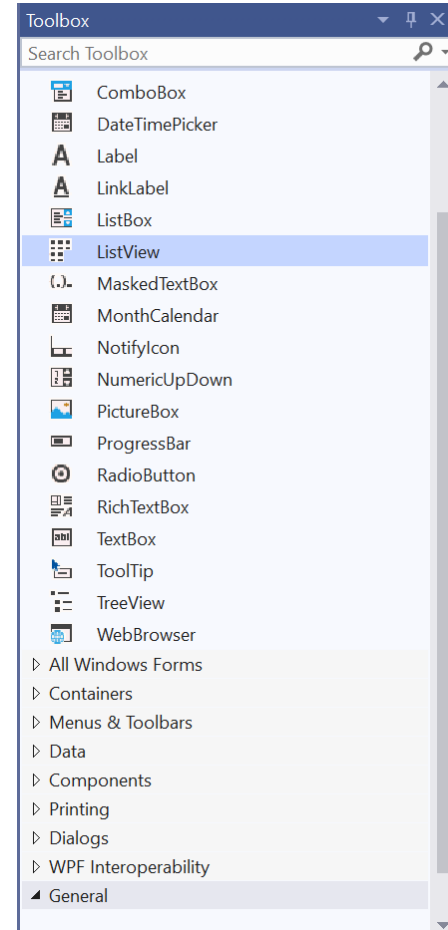


Common Controls

- **Label, LinkLabel**
- **TextBox**
- **Button**
- **GroupBox, Panel**
- **CheckBox, Radio Button**
- **ListBox, ComboBox**
- **Timer**
- **ImageList**
- **TabControl**
- **ListView**
- **TreeView**



— 상속 계층 구조

```
System.Object
System.MarshalByRefObject
System.ComponentModel.Component
System.Windows.Forms.Control           // 레벨
System.Windows.Forms.Label             // 링크레벨
System.Windows.Forms.LinkLabel
System.Windows.Forms.TextBoxBase
System.Windows.Forms.TextBox           // 텍스트박스
System.Windows.Forms.ButtonBase
System.Windows.Forms.Button            // 버튼
System.Windows.Forms.CheckBox          // 체크박스
System.Windows.Forms.RadioButton       // 라디오버튼
System.Windows.Forms.GroupBox          // 그룹상자
System.Windows.Forms.ScrollableControl
System.Windows.Forms.Panel             // 패널
System.Windows.Forms.ListControl
System.Windows.Forms.ComboBox          // 콤보 박스
System.Windows.Forms.ListBox           // 리스트 박스
System.Windows.Forms.CheckedListBox
```

Label

- 고정된 텍스트를 표시
- 사용자에게 제목이나 짧은 글을 표시하는 용도로 사용
- Image속성을 사용하여 이미지 표시 가능

주요 속성	설명
Text	레이블의 텍스트
TextAlign	레이블에서의 텍스트 정렬방식 지정 BottomCenter/BottomLeft/BottomRight ...
Image	레이블에 표시되는 이미지
BorderStyle	컨트롤의 테두리 Fixed3D/FixedSingle/None
Cursor	마우스 포인터가 컨트롤 위에 있을 때 표시되는 커서 설정

LinkLabel

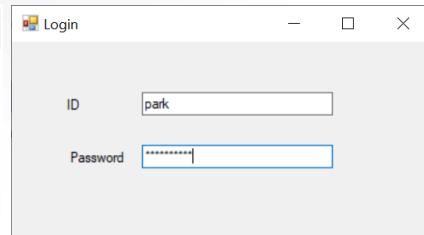
- 하이퍼링크를 표시할 수 있는 label
- 주요 속성
 - ✓ LinkArea, Links - 링크 표시
- 주요 이벤트
 - ✓ LinkClicked - 링크에 대한 처리
 - ✓ 하이퍼링크 클릭 시 해당 페이지를 웹 브라우저에 띄우는 처리 필요

```
LinkLabel linkLabel1 = new LinkLabel() ;  
linkLabel1.Text = "수업블로그로 직접 이동합니다.";  
linkLabel1.AutoSize = true;  
//라벨의 텍스트에서 링크로 처리할 영역 지정(0위치에서 크기 5영역)  
linkLabel1.LinkArea = new System.Windows.Forms.LinkArea(0, 5);  
//링크를 클릭했을 때 처리할 이벤트 핸들러 등록  
linkLabel1.LinkClicked +=  
    new linkLabelLinkClickedEventHandler( linkLabel1_LinkClicked);
```

TextBox

- 텍스트를 입력하고 표시하는 데 사용되는 컨트롤

주요 속성	설명
ReadOnly	True로 하여 출력전용으로 사용할 수 있음
Multiline, ScrollBars	여러 줄의 텍스트를 표시하거나 입력
MaxLength	TextBox에 입력되는 텍스트의 길이를 제한
PasswordChar	입력문자 표시를 숨김
CharacterCasing	대문자 또는 소문자로 선택가능 Normal/Upper/Lower
AcceptsTab, AcceptsReturn	제어키 사용 여부
주요 이벤트	설명
TextChanged	사용자가 키를 누를 때마다 발생



RichTextBox

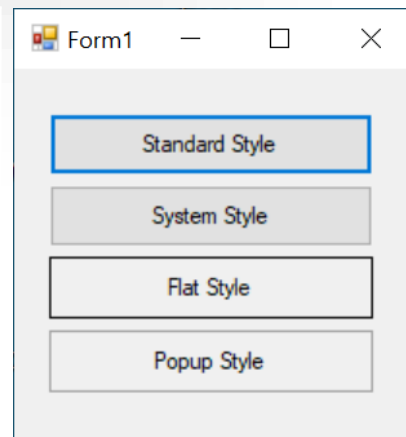
- 좀더 확장된 텍스트 컨트롤
- 텍스트박스베이스(TextBoxBase)라는 중간 클래스에서 파생
- 텍스트 선택, 잘라내기, 복사, 클립보드로부터 붙여넣기, 다양한 이벤트 등의 기능 제공

주요 속성	설명
SelectedText	사용자가 마우스로 선택한 텍스트 영역

Button

- 사용자로부터 특정 동작을 수행하도록 하는 컨트롤
- 단추에 포커스가 있는 경우 마우스, Enter 키 또는 스페이스바를 사용하여 클릭가능

주요 속성	설명
FlatStyle	단추의 모양을 변경 Standard/System/Flat/Popup/
Parent	컨트롤의 부모 컨테이너를 가져오거나 설정
FindForm()	컨트롤이 들어 있는 폼을 검색 Control p = btnParent.Parent; Form f = btnParent.FindForm();
주요 이벤트	설명
Click	사용자가 버튼을 누를 때 발생



Button

//LinkLabel, Label, TextBox, Button 생성

```
this.linkLabel1 = new System.Windows.Forms.LinkLabel();
```

```
this.textBox1 = new System.Windows.Forms.TextBox();
```

```
this.button1 = new System.Windows.Forms.Button();
```

```
this.label1 = new System.Windows.Forms.Label();
```

```
this.label2 = new System.Windows.Forms.Label();
```

```
this.label3 = new System.Windows.Forms.Label();
```

//linkLabel1 속성 및 이벤트 설정

```
this.linkLabel1.AutoSize = true;
```

```
this.linkLabel1.Location = new System.Drawing.Point(36, 100);
```

```
this.linkLabel1.Name = "linkLabel1";
```

```
this.linkLabel1.Size = new System.Drawing.Size(287, 12);
```

```
this.linkLabel1.TabIndex = 0;
```

```
this.linkLabel1.TabStop = true;
```

```
this.linkLabel1.Text = "HCI Programming 수업블로그로 직접 이동합니다.";
```

```
this.linkLabel1.LinkClicked += new  
System.Windows.Forms.LinkLabelLinkClickedEventHandler(this.linkLabel1_LinkClicked);
```

//button1 속성 및 이벤트 설정

```
this.button1.Location = new System.Drawing.Point(260, 65);
```

```
this.button1.Name = "button1";
```

```
this.button1.Size = new System.Drawing.Size(75, 23);
```

```
this.button1.TabIndex = 2;
```

```
this.button1.Text = "이동";
```

```
this.button1.UseVisualStyleBackColor = true;
```

```
this.button1.Click += new System.EventHandler(this.button1_Click);
```

Button

//웹 페이지 링크레이블 클릭시 수업블로그로 이동

```
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) {
    try {
        linkLabel1.LinkVisited = true;
        // call the Process.Start method to open the default browser with a URL
        System.Diagnostics.Process.Start("http://dis.dankook.ac.kr/lectures/hci21/");
    }
    catch (Exception ex) {
        MessageBox.Show("Unable to open link that was clicked.");
    }
}
```

//버튼 클릭시 textBox1에 입력한 웹 페이지 링크로 이동

```
private void button1_Click(object sender, EventArgs e) {
    try {
        // call the Process to open the default browser with a user-input URL
        System.Diagnostics.Process ieProcess = new System.Diagnostics.Process();
        ieProcess.StartInfo.FileName = this.textBox1.Text;
        ieProcess.Start();
    }
    catch (Exception ex) {
        MessageBox.Show("Unable to open link.");
    }
}
```

— 그룹상자

- 폼에 배치된 컨트롤들을 그룹화하기 위해 사용
- 라디오버튼이나 체크버튼 같이 관련 있는 컨트롤들을 기능별로 분류하여 시각적으로 표현할 때 사용

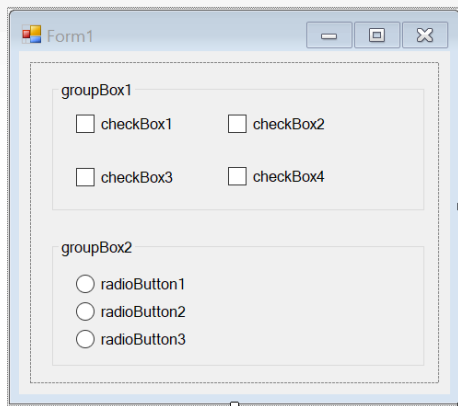
— 컨트롤 그룹생성방법

- 폼에 GroupBox 컨트롤을 배치
- 그룹 내에 포함할 컨트롤들을 그룹 상자 안에 배치
- 기존 컨트롤을 그룹 상자에 포함하려면 포함할 모든 컨트롤을 선택하여 클립보드에 잘라 낸 다음 GroupBox 컨트롤을 선택하여 그룹 상자에 붙여 넣기 또는 그룹 상자에 끌어 오기
- 그룹 상자의 Text 속성에 적절한 캡션을 설정

패널

- 컨트롤 그룹화에 이용되는 컨트롤
- 캡션지정을 위한 Text속성이 없으므로 캡션기능이 필요한 그룹 설정 시 그룹상자 (GroupBox) 컨트롤 사용
- 주요 속성
 - ✓ **AutoScroll** - ScrollableControl클래스에서 파생되므로 AutoScroll 속성을 사용하여 Panel 컨트롤에서 스크롤 막대 활성화

GroupBox



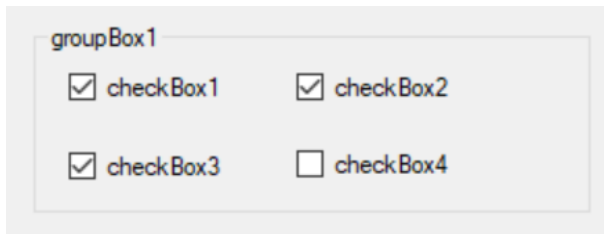
Panel

— 체크박스

- 여러 개의 확인란을 선택할 수 있는 컨트롤

주요 속성	설명
Appearance	체크박스를 일반적으로 표시할지 또는 버튼 모양으로 표시할지 여부를 결정
Checked	True/False
CheckState	Checked/Indeterminate/Unchecked

주요 이벤트	설명
Click	컨트롤을 클릭하면 Checked 속성이 True로 설정되고 Click 이벤트 처리
CheckedChanged	Checked 속성이 값이 변경되면 발생



— 라디오 버튼

- 사용자가 선택 그룹에서 단일 옵션을 선택해야 할 경우에 사용

주요 속성	설명
Text	라디오 버튼의 설명 및 선택키 표시 (선택키를 &과 문자의 조합으로 설정하면 실행 시 ALT와 함께 써서 Click 기능)
Checked	True/False
CheckState	Checked/Unchecked
AutoCheck	속성이 True(기본값)로 설정된 라디오 버튼을 선택하면 해당 그룹의 다른 모든 선택표시가 자동으로 제거

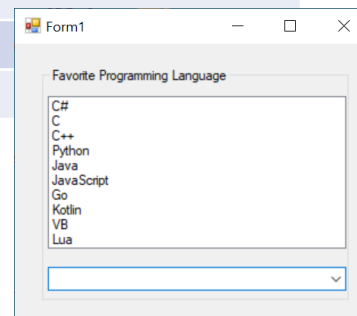
주요 이벤트	설명
Click	Checked 속성이 True로 설정되고 Click 이벤트 처리기가 호출
CheckedChanged	Checked 속성이 값이 변경되면 발생



ListBox

- 다수의 항목에서 단일/다중 선택할 수 있는 항목 목록을 표시

주요 속성	설명
SelectionMode	단일 또는 다중 선택 옵션 MultiExtended/MultiSimple/None/One
MultiColumn	리스트 박스의 항목들을 열 형식으로 표시 여부
Items	리스트 박스의 항목들에 대한 컬렉션. Add/AddRange/Insert/Remove/Clear 메소드로 항목을 추가, 제거
SelectedItem	현재 선택된 항목의 텍스트 (선택이 없을 경우 null)
SelectedIndex	현재 선택된 항목의 인덱스 (선택이 없을 경우 -1)
SelectedItems	다중 선택된 항목들에 대한 컬렉션
SelectedIndices	다중 선택된 항목들에 대한 인덱스들에 대한 컬렉션
Sorted	목록의 내용을 알파벳 순서대로 정렬



— 주요 이벤트

- **Click** - 항목을 클릭할 때 발생
- **DoubleClick** - 항목을 더블 클릭할 때 발생
- **SelectedValueChanged** - ListBox의 항목선택을 변경하면 SelectedValue속성이 변경되고, SelectedValueChanged 이벤트 발생
- **SelectedIndexChanged** - SelectedIndex 속성이 변경될 때 발생
 - ✓ ListBox의 선택된 인덱스가 변경된 시기를 확인

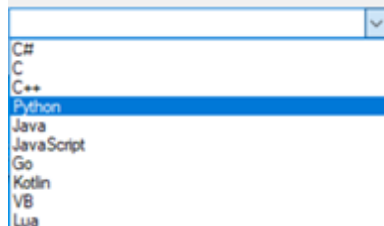
— 주요 메소드

- **public void SetSelected (int index, bool value)**
 - ✓ Listbox의 지정된 항목을 선택하거나 선택 취소
- **public bool GetSelected (int index)**
 - ✓ 지정된 항목이 선택되어 있는지 여부를 나타내는 값을 반환
- **public int FindString (string s)**
public int FindString (string s, int startIndex)
FindStringExact()
 - ✓ Listbox에 포함된 항목 검색

ComboBox

- 목록선택과 텍스트 편집기능을 포함한 형태 (ListBox+ TextBox)

주요 속성	설명
ComboBoxStyle	콤보 박스 스타일 DropDown (기본스타일)/DropDownList/Simple
SelectedItem	현재 선택된 항목의 텍스트 (선택이 없을 경우 null)
SelectedIndex	현재 선택된 항목의 인덱스 (선택이 없을 경우 -1)
주요 이벤트	설명
TextChanged	Text 속성이 변경되면 발생/리스트의 항목을 선택하거나 텍스트 필드에 값을 입력할 때 발생
SelectedIndexChanged	SelectedIndex 속성이 변경되면 발생
SelectedValueChanged	SelectedValue 속성이 변경되면 발생
SelectionChangeCommitted	선택된 항목이 변경되고 해당 변경 내용이 ComboBox에 표시될 때 발생



ComboBox

// 폼에 각종 컨트롤 생성

this.panel1 = new System.Windows.Forms.Panel(); **// 패널1**

this.groupBox1 = new System.Windows.Forms.GroupBox(); **// 그룹박스1**

this.radioButton1 = new System.Windows.Forms.RadioButton(); **// 라디오버튼1**

this.radioButton2 = new System.Windows.Forms.RadioButton();

this.groupBox2 = new System.Windows.Forms.GroupBox();

this.checkBox1 = new System.Windows.Forms.CheckBox();

this.checkBox2 = new System.Windows.Forms.CheckBox();

this.checkBox3 = new System.Windows.Forms.CheckBox();

this.checkBox4 = new System.Windows.Forms.CheckBox();

this.groupBox3 = new System.Windows.Forms.GroupBox();

this.listBox1 = new System.Windows.Forms.ListBox();

this.comboBox1 = new System.Windows.Forms.ComboBox();

// 중간 생략..

// panel1

this.panel1.Controls.Add(this.groupBox3); **// 패널에 그룹박스1 추가**

this.panel1.Controls.Add(this.groupBox2); **// 패널에 그룹박스2 추가**

this.panel1.Controls.Add(this.groupBox1); **// 패널에 그룹박스3 추가**

this.panel1.Location = new System.Drawing.Point(28, 65);

this.panel1.Name = "panel1";

this.panel1.Size = new System.Drawing.Size(225, 335);

this.panel1.TabIndex = 2;

// Form1

this.Controls.Add(this.panel1); **// 폼에 패널추가**

CheckBox

```
// CheckBox
this.checkBox4 = new System.Windows.Forms.CheckBox();
this.checkBox3 = new System.Windows.Forms.CheckBox();
this.checkBox2 = new System.Windows.Forms.CheckBox();
this.checkBox1 = new System.Windows.Forms.CheckBox();

// checkBox1
this.checkBox1.AutoSize = true;
this.checkBox1.Checked = true;
this.checkBox1.CheckState = System.Windows.Forms.CheckState.Checked;
this.checkBox1.Location = new System.Drawing.Point(8, 22);
this.checkBox1.Name = "checkBox1";
this.checkBox1.Size = new System.Drawing.Size(86, 16);
this.checkBox1.TabIndex = 0;
this.checkBox1.Text = "checkBox1";
this.checkBox1.UseVisualStyleBackColor = true;
this.checkBox1.CheckedChanged += new System.EventHandler(checkBox_CheckedChanged);

// checkBox2
this.checkBox2.Text = "checkBox2";
this.checkBox2.CheckedChanged += new System.EventHandler(checkBox_CheckedChanged);

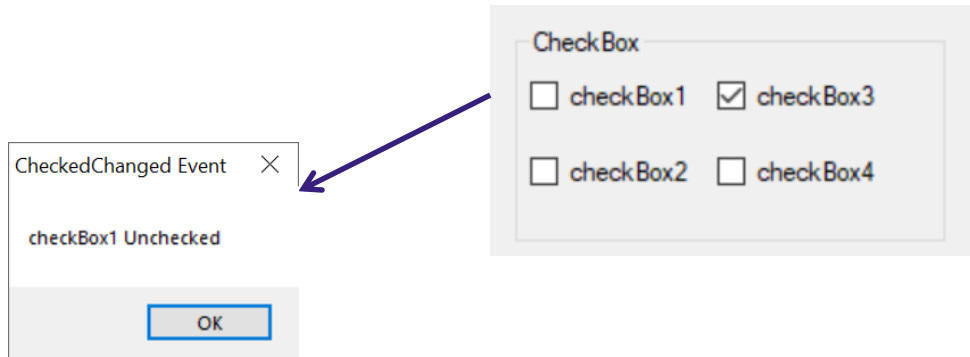
// checkBox3
this.checkBox3.Text = "checkBox3";
this.checkBox3.CheckedChanged += new System.EventHandler(checkBox_CheckedChanged);

// checkBox4
this.checkBox4.Text = "checkBox4";
this.checkBox4.CheckedChanged += new System.EventHandler(checkBox_CheckedChanged);
```

CheckBox

// checkBox_CheckedChanged 이벤트 처리

```
private void checkBox_CheckedChanged(object sender, EventArgs e)
{
    CheckBox cb = (CheckBox) sender;
    MessageBox.Show(cb.Text + " " + cb.CheckState, "CheckedChanged Event");
}
```



Radio Button

// RadioButton

```
System.Windows.Forms.RadioButton radioButton1;  
System.Windows.Forms.RadioButton radioButton2;  
System.Windows.Forms.RadioButton radioButton3;
```

// radioButton1

```
radioButton1.Text = "radioButton1(&A)"; //선택키 설정(Alt+A)  
radioButton1.Click += new System.EventHandler(this.radioButton_Click);
```

// radioButton2

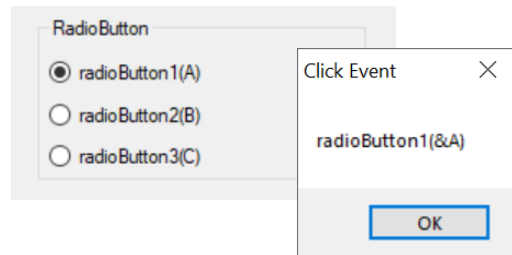
```
radioButton2.Checked = true; //초기 체크상태  
radioButton2.Text = "radioButton2(&B)";  
radioButton2.Click += new System.EventHandler(this.radioButton_Click);
```

// radioButton3

```
radioButton3.Text = "radioButton3(&C)";  
radioButton3.Click += new System.EventHandler(this.radioButton_Click);
```

// RadioButton Click 이벤트 처리

```
private void radioButton_Click(object sender, EventArgs e)  
{  
    RadioButton rb = (RadioButton)sender;  
    MessageBox.Show(rb.Text, "Click Event");  
}
```



Listbox

//ListBox

```
System.Windows.Forms.ListBox listBox1;
```

// listBox1 디자인에서 추가된 항목들

```
listBox1.Items.AddRange(new object[] { "봄", "여름", "가을", "겨울"});
```

```
listBox1.Sorted = true; //정렬
```

```
listBox1.SelectionMode = System.Windows.Forms.SelectionMode.MultiExtended; //다중선택
```

//선택항목이 변경될 때 처리할 이벤트 핸들러 등록

```
listBox1.SelectedIndexChanged += new
```

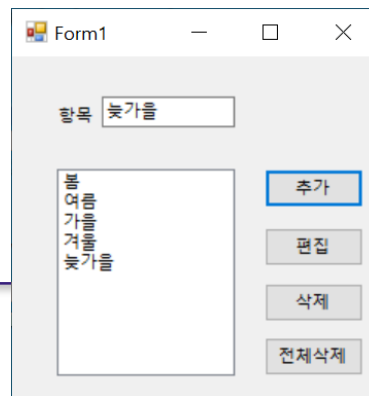
```
    System.EventHandler(this.listBox1_SelectedIndexChanged);
```

//항목을 더블 클릭할 때 처리할 이벤트 핸들러 등록

```
listBox1.DoubleClick += new System.EventHandler(this.listBox1_DoubleClick);
```

//listBox1의 항목을 더블 클릭하면 선택된 항목을 삭제

```
void listBox1_DoubleClick(object sender, EventArgs e) {  
    if (listBox1.SelectedItem != null) {  
        listBox1.Items.Remove(listBox1.SelectedItem);  
    }  
}
```



Listbox

//추가버튼을 누르면 리스트에 항목추가

```
private void add_Click(object sender, EventArgs e) {  
    string item = textBox1.Text;  
    listBox1.Items.Add(item);  
}
```

//편집버튼을 누르면 기존의 것을 지우고 새로운 것을 삽입

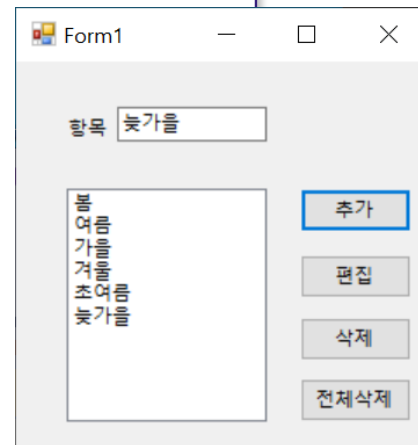
```
private void edit_Click(object sender, EventArgs e){  
    if(listBox1.SelectedIndex != -1){  
        string item = textBox1.Text;  
        listBox1.Items.Insert(listBox1.SelectedIndex, item);  
        listBox1.Items.RemoveAt(listBox1.SelectedIndex);  
    }  
}
```

//삭제버튼을 누르면 선택된 항목을 삭제

```
private void remove_Click(object sender, EventArgs e) {  
    if (listBox1.SelectedItem != null) {  
        listBox1.Items.Remove(listBox1.SelectedItem);  
    }  
}
```

//전체삭제버튼을 누르면 모든 항목을 삭제

```
private void removeall_Click(object sender, EventArgs e) {  
    listBox1.Items.Clear();  
}
```

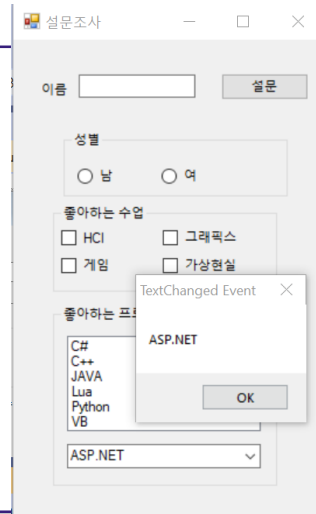


ComboBox

// ComboBox

```
this.comboBox1.Items.AddRange(new object[] {  
    "ASP.NET",  
    "ASP",  
    "JSP",  
    "PHP"});  
  
this.comboBox1.SelectedIndexChanged +=  
    new System.EventHandler(this.comboBox1_SelectedIndexChanged);  
  
this.comboBox1.TextChanged +=  
    new System.EventHandler(this.comboBox1_TextChanged);
```

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e) {  
    MessageBox.Show(comboBox1.SelectedItem + "(" +  
        comboBox1.SelectedIndex + ")",  
        "SelectedIndexChanged Event");  
}  
  
private void comboBox1_TextChanged(object sender, EventArgs e) {  
    MessageBox.Show(comboBox1.Text, "TextChanged Event");  
}
```



— 상속 계층 구조

System.Object	
System.MarshalByRefObject	
System.ComponentModel.Component	
System.Windows.Forms.Timer	// 타이머
System.Windows.Forms.ImageList	// 이미지 리스트
System.Windows.Forms.Control	
System.Windows.Forms.TabControl	// 탭 컨트롤
System.Windows.Forms.ListView	// 리스트 뷰
System.Windows.Forms.TreeView	// 트리 뷰
System.Windows.Forms.Toolbar	// 툴바
System.Windows.Forms.ScrollBar	// 스크롤바
System.Windows.Forms.ProgressBar	// 프로그래스바
System.Windows.Forms.PictureBox	// 그림 상자
System.Windows.Forms.MonthCalendar	// 달력

Timer

- 정해진 시간 간격마다 이벤트를 발생시키는 컨트롤
- 용도 - Multitasking, 변화되는 상태 표현 (진행상태), 자동저장기능, 애니메이션

주요 속성	설명
Enabled	타이머 실행 여부를 설정
Interval	타이머 이벤트 발생 간격(시간) (단위: millisecond)
주요 메소드	설명
Start()/Stop()	타이머를 시작/중지
주요 이벤트	설명
Tick	지정된 타이머 간격이 경과되고 타이머를 사용할 수 있을 때 발생

PictureBox

- 이미지를 표시하는데 사용

주요 속성	설명
Image	표시될 이미지 지정
SizeMode	이미지 크기와 위치 조절 Normal/CenterImage/StretchImage/AutoSize/Zoom
주요 이벤트	설명
Click	그림상자 클릭 시 발생

PictureBox Timer

//PictureBox

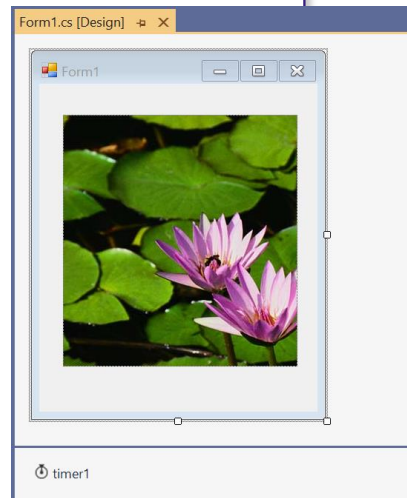
```
this.pictureBox1.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));  
this.pictureBox1.Location = new System.Drawing.Point(25, 25);  
this.pictureBox1.Name = "pictureBox1";  
this.pictureBox1.Size = new System.Drawing.Size(246, 204);  
this.pictureBox1.TabIndex = 0;  
this.pictureBox1.TabStop = false;
```

//Timer

```
this.timer1.Enabled = true;  
this.timer1.Interval = 1000;  
this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
```

//Timer Tick 이벤트 핸들러

```
private void timer1_Tick(object sender, EventArgs e) {  
    if (forward)  
        alpha -= 0.01;  
    else  
        alpha += 0.01;  
    if (alpha <= 0.5 || alpha >= 1.0) // alpha: 0.5 ~ 1.0  
        forward = !forward;  
    this.Opacity = alpha;  
    System.Diagnostics.Trace.WriteLine("Debug: alpha=" + alpha);  
}
```



— 이미지 목록

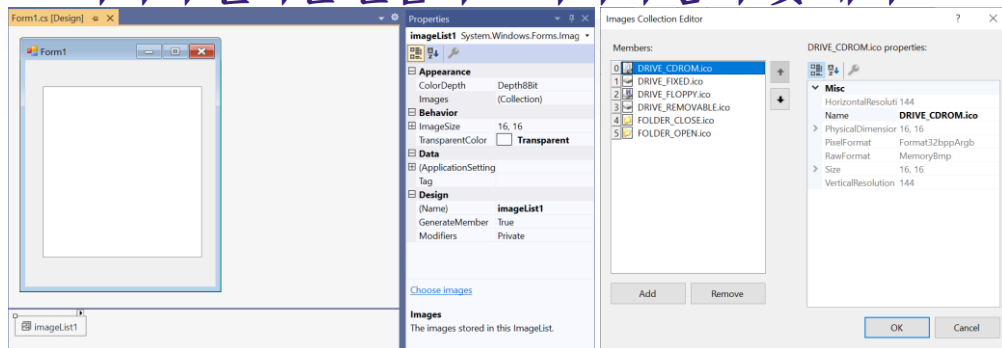
- 동일한 크기의 이미지 집합
- 다른 컨트롤에서 표시할 수 있는 이미지를 저장
- 이미지 목록을 사용하면 하나의 일관된 이미지 카탈로그에 대한 코드 작성 가능
- 컨트롤들 사이에서의 이미지 공유 가능
- 각각의 이미지를 배열 항목처럼 취급하여 0부터 시작하는 인덱스로 참조

— 이미지 목록과 연결할 수 있는 컨트롤

- ListView, TreeView, ToolBar, TabControl, Button, CheckBox, RadioButton, Label

주요 속성

- **Images** - 이미지 리스트의 이미지들을 관리하는 컬렉션으로 이미지 컬렉션 편집기로 이미지 등록 및 제거



// ImageList 생성 및 속성 설정 예

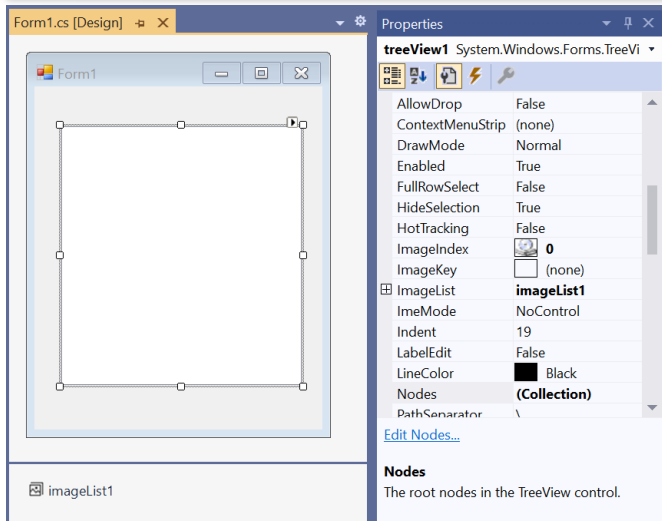
```
System.Windows.Forms.ImageList imageList1 =  
    new System.Windows.Forms.ImageList(this.components);  
imageList1.ImageStream = ((System.Windows.Forms.ImageListStreamer)  
    (resources.GetObject("imageList1.ImageStream")));  
imageList1.TransparentColor = System.Drawing.Color.Transparent;  
imageList1.Images.SetKeyName(0, "textdoc.ico");  
imageList1.Images.SetKeyName(1, "inifile.ico");
```

— 컨트롤에 이미지 리스트 적용

- 컨트롤의 ImageList 속성에 배치된 ImageList 컨트롤을 연결
- ImageIndex 또는 ImageKey에 연결할 이미지 선택

//이미지리스트의 0번째 인덱스에 등록 된 이미지를 트리뷰의 이미지로 설정

```
treeView1.ImageIndex = 0;
```



TabControl

- 여러 페이지로 구성된 대화 상자 구성에 이용

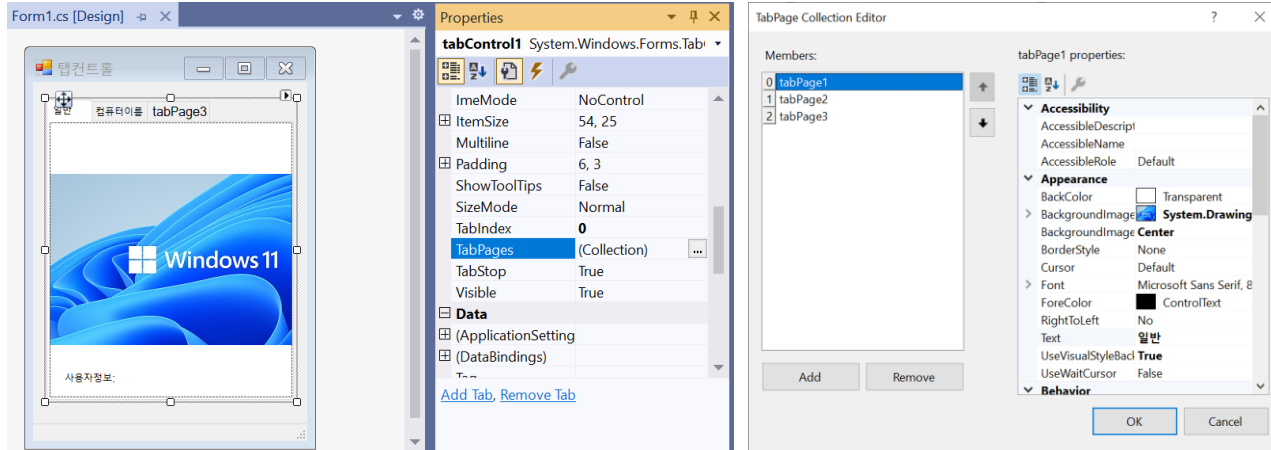
주요 속성	설명
TabPage	탭 페이지(Tab Page) 들의 컬렉션
SelectedIndex	탭 페이지 선택 인덱스 확인 및 변경
SelectedTab	탭 페이지 선택 확인 및 변경

주요 이벤트	설명
Click	탭을 클릭할 때 발생
Selected	탭을 선택할 때 발생

주요 메소드	설명
SelectTab(Int32)	지정된 인덱스의 탭을 현재 탭으로 설정
SelectTab(String)	지정된 이름의 탭을 현재 탭으로 설정
SelectTab(TabPage)	지정된 탭 페이지를 현재 탭으로 설정

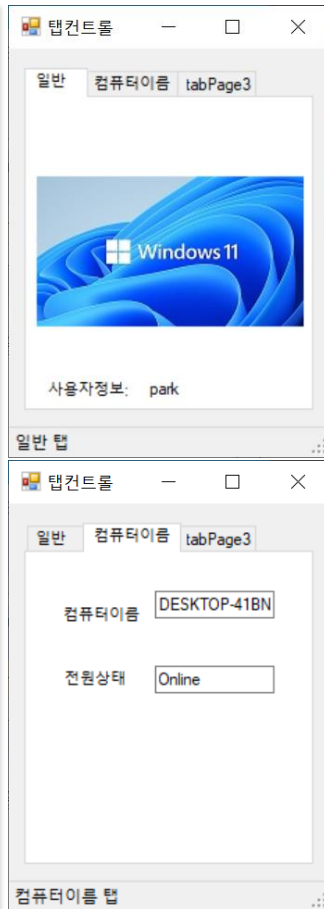
TabPage 속성 편집기

— TabPages를 구성하는 TabPage의 속성을 편집



TabControl

```
// TabControl
this.tabControl1.Controls.Add(this.tabPage1);
this.tabControl1.Controls.Add(this.tabPage2);
this.tabControl1.Name = "tabControl1";
this.tabControl1.Selected += new
    System.Windows.Forms.TabControlEventHandler(this.tabControl1_Selected);
// tabPage1
this.tabPage1.BackgroundImage =
    ((System.Drawing.Image)(resources.GetObject("tabPage1.BackgroundImage")));
this.tabPage1.Controls.Add(this.label2);
this.tabPage1.Controls.Add(this.label1);
this.tabPage1.Text = "일반";
// tabPage2
this.tabPage2.Controls.Add(this.textBox2);
this.tabPage2.Controls.Add(this.textBox1);
this.tabPage2.Controls.Add(this.label4);
this.tabPage2.Controls.Add(this.label3);
this.tabPage2.Text = "컴퓨터이름";
// Selected 이벤트 핸들러
private void tabControl1_Selected(object sender, EventArgs e) {
    if (tabControl1.SelectedTab.Text == "일반") {
        this.label2.Text = SystemInformation.UserName;
        this.toolStripStatusLabel1.Text = "일반 탭";
    } else if (tabControl1.SelectedTab.Text == "컴퓨터이름") {
        // 중간생략
    }
}
```



— ListView

- 아이콘과 함께 항목 목록을 표시하는 컨트롤
- 각 목록 항목에 대해 여러 개의 열 표시 가능

— ListView의 View 속성

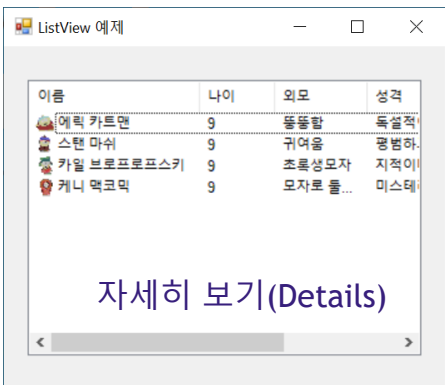
- 자세히 보기
- 큰 아이콘 보기
- 작은 아이콘 보기
- 리스트 보기
- 타일 보기

Listview의 View 속성

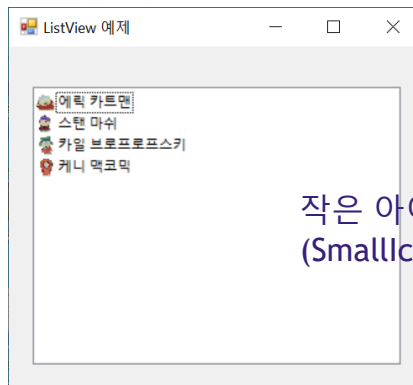
큰 아이콘 보기
(LargeIcon)



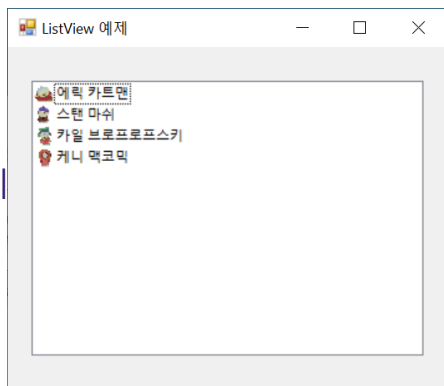
자세히 보기(Details)



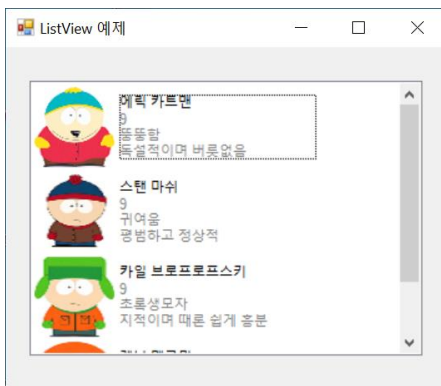
작은 아이콘 보기
(SmallIcon)



리스트 보기
(List)



타일
보기
(Tile)



— ListViewItem

- ListView의 Items 속성에 추가되는 항목

— ListViewSubItem

- ListView의 각 Item에 추가되는 하위항목

ListviewItem

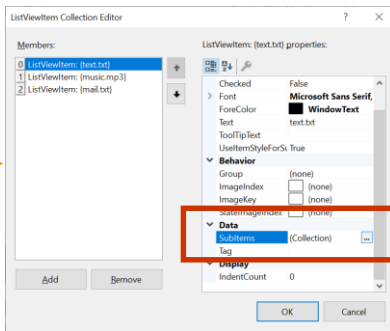
- Listview의 Items 속성에 추가되는 항목

ListviewSubItem

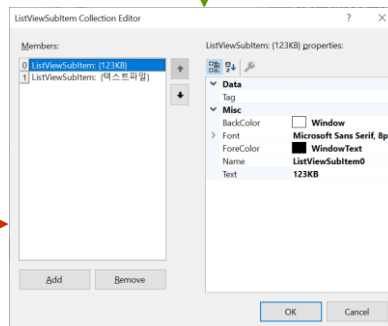
- Listview의 각 Item에 추가되는 하위항목

파일명	크기	설명
text.txt	123KB	텍스트파일
music.mp3	4.2MB	음악파일
mail.txt	23KB	편지파일

ListviewItem



ListviewSubItem

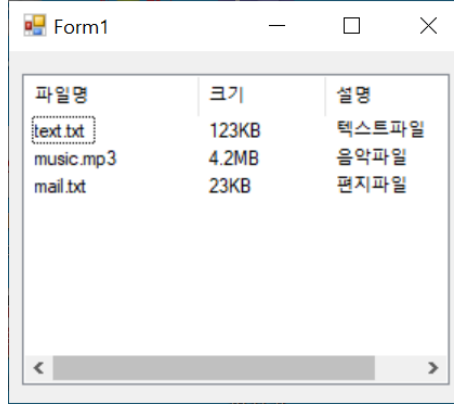
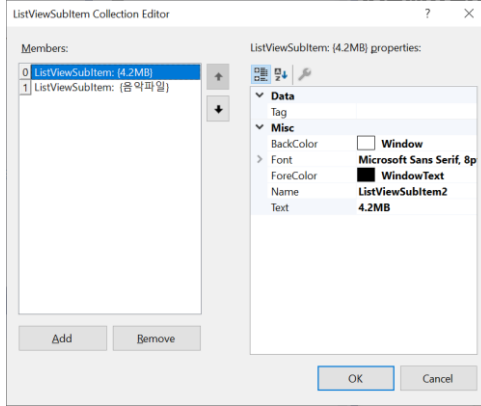
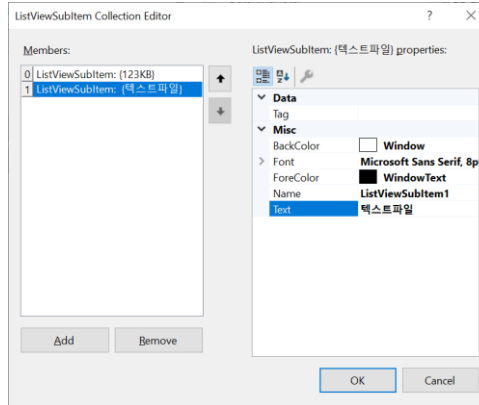
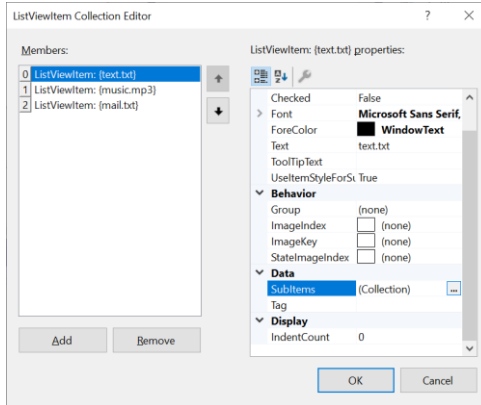


Items Collection

The screenshot illustrates the process of configuring a List View control in Visual Studio. It shows the following components:

- Toolbox:** The **List View** control is selected.
- Form1:** The List View control is placed on the form.
- List View Items Collection Editor:** This dialog shows the **Members** list with three items:
 - 0 ListViewItem: [text.txt]
 - 1 ListViewItem: [music.mp3]
 - 2 ListViewItem: []The **Properties** window for the selected item shows:
 - Appearance:** BackColor (Window), Checked (False), Font (Microsoft Sans Serif), ForeColor (WindowText), Text (text.txt), ToolTipText, UseltemStyleForSt (True).
 - Behavior:** Group (none), ImageIndex (none), ImageKey (none), StateImageIndex (none).
 - Data:** SubItems (Collection), Tag.
 - Display:** IndentCount (0).
- Form1.cs [Design]:** The design view shows the List View containing the text: `text.txt music... mail.txt`.
- Properties:** The **listView1** properties window shows the **View** property set to **LargeIcon**.

Subitems Collection



ColumnHeader

- 리스트 뷰의 자세히 보기에서 항목의 정보(열 머리글)를 표현하는 집합

• 리스트 뷰의 Columns 속성 참조 시용하여 추가

// ColumnHeader 생성 및 ListView의 Columns 속성 설정 예

```
System.Windows.Forms.ColumnHeader columnHeader1 =  
    new System.Windows.Forms.ColumnHeader();  
columnHeader1.Text = "파일명";  
columnHeader1.Width = 100;  
//...  
listView1.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] {  
    columnHeader1, columnHeader2, columnHeader3});
```

List View

- Columns Collection

The screenshot displays three windows from Visual Studio:

- Form1.cs [Design]:** Shows a ListView control with three columns. The data is as follows:

파일명	크기	설명
text.txt	123KB	텍스트파일
music.mp3	4.2MB	음악파일
mail.txt	23KB	편지파일
- Properties:** Shows the properties for the `listView1` control. The **Columns** property is highlighted, indicating it is a collection.

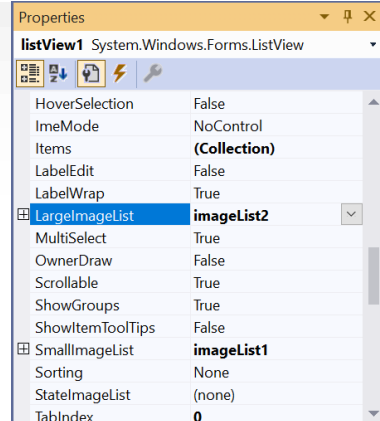
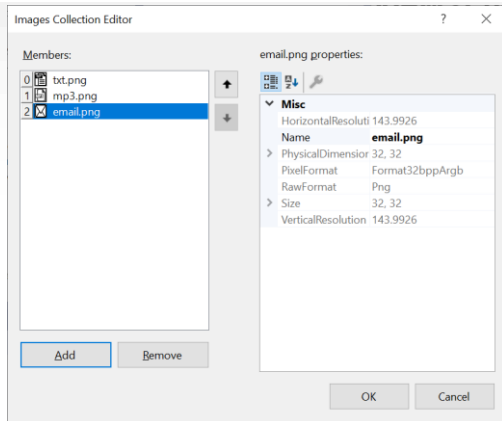
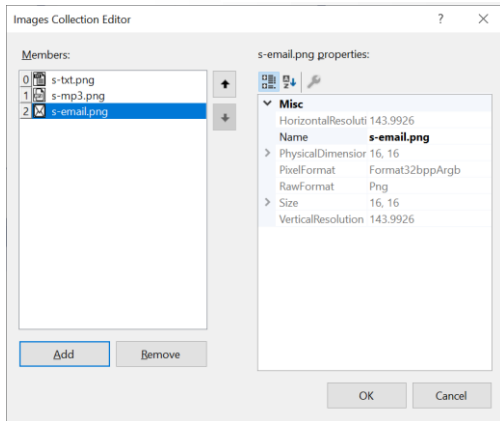
AllowDrop	False
AutoArrange	True
Columns	(Collection)
ContextMenuSt	(none)
Enabled	True
Groups	(Collection)
HeaderStyle	Clickable
HideSelection	False
HotTracking	False
HoverSelection	False
ImeMode	NoControl
Items	(Collection)
- ColumnHeader Collection Editor:** Shows the collection of column headers. The `columnHeader1` property is selected, and its details are shown on the right.

columnHeader1 properties:	
Behavior	DisplayIndex: 0
Data	(ApplicationSetting): Tag
Design	(Name): columnHeader1 GenerateMember: True Modifiers: Private
Misc	ImageIndex: (none) ImageKey: (none) Text: 파일명 TextAlign: Left Width: 60

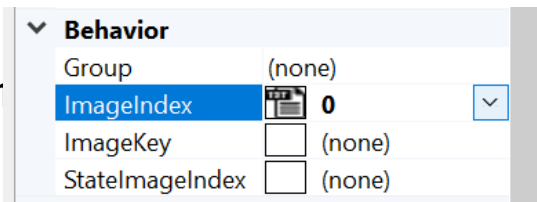
Listview - Add Icons

— **Listview**의 아이콘 이미지 연결을 위하여 **ImageList** 컨트롤 사용

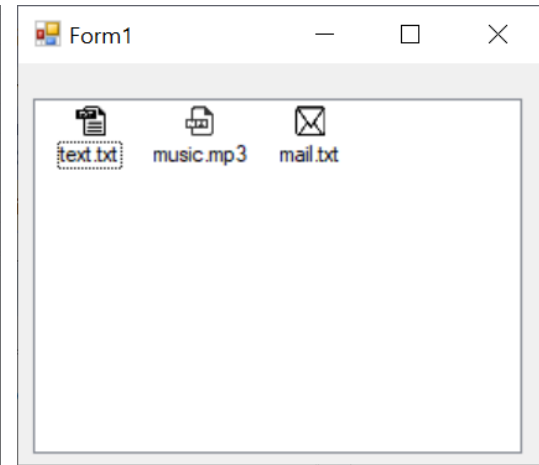
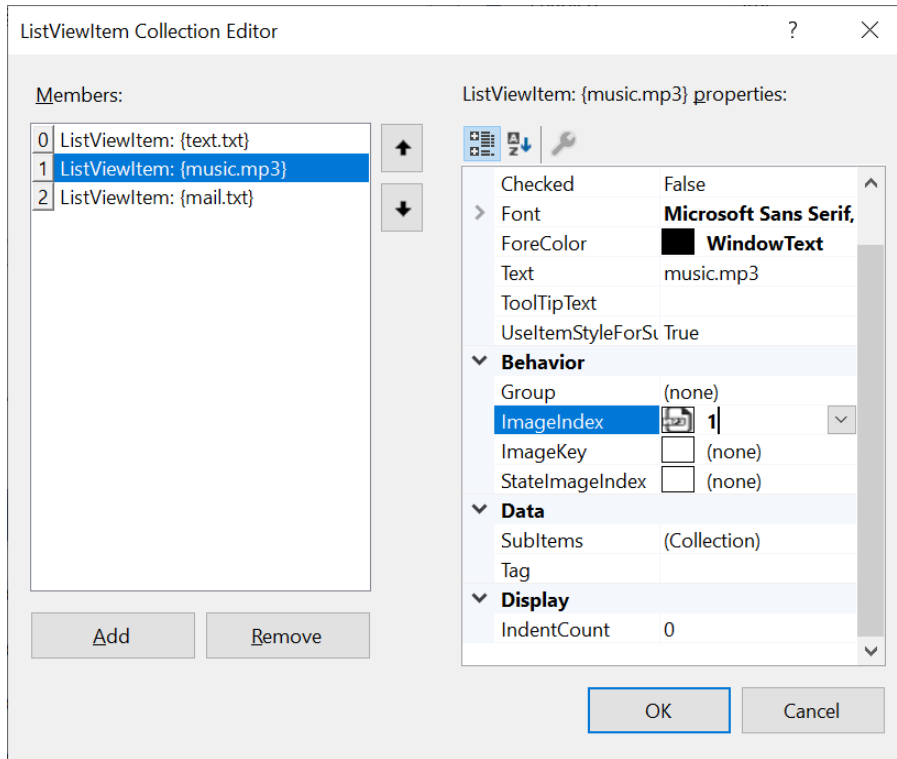
- **SmallImageList** - 작은 아이콘 이미지(16X16)
- **LargeImageList** - 큰 아이콘 이미지(32X32)



— **Listview item**의 **ImageList**
속성 설정



ListView - Add Icons



List View

주요 속성	설명
FullRowSelect	해당 항목의 모든 하위 항목 선택 여부 설정
List View.GridLines	컨트롤에서 항목과 하위항목이 포함된 행과 열 사이에 모눈선을 나타낼지 여부를 설정
AllowColumnReorder	컨트롤에서 열 머리글을 끌어 열의 순서 변경 여부 설정
LabelEdit	항목의 레이블 편집 여부 설정 BeforeLabelEdit/AfterLabelEdit 이벤트를 사용하여 항목테스트를 편집하기 전/후에 작업 수행
CheckBoxes	컨트롤에 있는 각 항목 옆에 확인란을 표시할지 여부 설정
CheckedItems	확인란이 선택된 모든 항목을 접근하기 위한 컬렉션 (List View.CheckedIndexCollection)
CheckedIndices	확인란이 선택된 모든 항목의 인덱스 접근

— 주요 이벤트

- **ItemSelectionChanged** – 항목의 선택 상태가 변경되면 발생
- **ColumnClick** – 사용자가 ListView 컨트롤에서 열 머리글을 클릭할 때 발생
- **ItemCheck** – 항목 확인란 선택 시 발생
- **DoubleClick** – 항목을 더블 클릭 시 발생

```
private void listView1_ColumnClick(object sender, ColumnClickEventArgs e) {  
    int index = e.Column; //선택된 컬럼의 인덱스  
    string str = listView1.Columns[index].Text; //선택된 컬럼의 문자열 확인  
    MessageBox.Show(str, "ColumnClick Event");  
}  
private void listView1_ItemSelectionChanged(object sender,  
    ListViewItemSelectionChangedEventArgs e) {  
    string str = e.Item.Text + ", selected:" + e.IsSelected; //변경된 항목의 상태 확인  
    MessageBox.Show(str, "ItemSelectionChanged Event");  
}  
private void listView1_DoubleClick(object sender, EventArgs e) {  
    listView1.Items.Remove(listView1.FocusedItem); // 현재 선택된 항목 삭제  
}
```

List View

//ListView 생성

```
private System.Windows.Forms.ListView listView1 = new System.Windows.Forms.ListView();
```

// ListView의 ListViewItem 생성

```
System.Windows.Forms.ListViewItem listViewItem1 =
```

```
    new System.Windows.Forms.ListViewItem(new string[] { "text.txt", "123KB", "텍스트파일"}, 0);
```

```
System.Windows.Forms.ListViewItem listViewItem2 =
```

```
    new System.Windows.Forms.ListViewItem(new string[] { "music.mp3", "4.2MB", "음악파일"}, 1);
```

```
System.Windows.Forms.ListViewItem listViewItem3 =
```

```
    new System.Windows.Forms.ListViewItem(new string[] { "mail.txt", "23KB", "편지파일"}, 2);
```

//ListView의 ColumnHeader 생성

```
listView1.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] {
```

```
    columnHeader1, columnHeader2, columnHeader3});
```

//ListView의 items에 생성한 ListViewItem들을 추가

```
listView1.Items.AddRange(new System.Windows.Forms.ListViewItem[] {
```

```
    listViewItem1, listViewItem2, listViewItem3});
```

//이미지리스트 연결

```
listView1.LargeImageList = imageListL;
```

```
listView1.SmallImageList = imageListS;
```

//ListView의 보기(View) 설정

```
listView1.View = System.Windows.Forms.View.Details;
```


ListView

//코드를 사용한 리스트뷰 항목 생성 및 속성 설정

//새로운 ListViewItem 생성

```
ListViewItem newItem = new ListViewItem();
```

```
newItem.ImageIndex = 0;
```

//이미지리스트의 인덱스 설정

```
newItem.Text = "addtest.txt";
```

//항목의 텍스트 설정

//새로운 ListViewItem 의 ListViewSubItem 생성 및 속성 설정

```
ListViewItem .ListViewSubItem newSubItem1 = new ListViewItem.ListViewSubItem();
```

```
ListViewItem .ListViewSubItem newSubItem2 = new ListViewItem.ListViewSubItem();
```

```
newSubItem1.Text = "50KB";
```

```
newSubItem2.Text = "추가연습 문서파일";
```

//새로운 ListViewItem 에 ListViewSubItem 추가

```
newItem.SubItems.Add (newSubItem1);
```

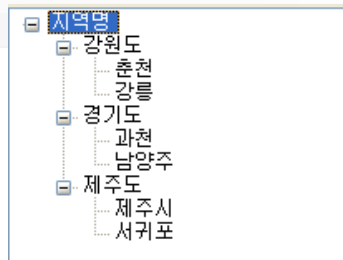
```
newItem.SubItems.Add (newSubItem2);
```

//ListView항목에 새로운 ListViewItem 추가

```
listView1.Items.Add(newItem);
```

TreeView

- 노드를 계층 구조 방식으로 표시하는 컨트롤
- Nodes 컬렉션: TreeView 컨트롤에 지정된 모든 TreeNode 개체들의 집합
- 루트 노드와 자식 노드로 구성
- 모든 노드는 자식 노드라고 하는 다른 노드 포함가능
- 노드를 확장 또는 축소 형태로 표시
 - ✓ Nodes 컬렉션에 있는 모든 자식 트리 노드 수준을 확장 : ExpandAll()
 - ✓ TreeNode 수준을 확장 : Expand()
 - ✓ TreeNode 수준을 축소 : Collapse()
 - ✓ 확장된 상태와 축소된 상태 사이에서 전환 : Toggle()

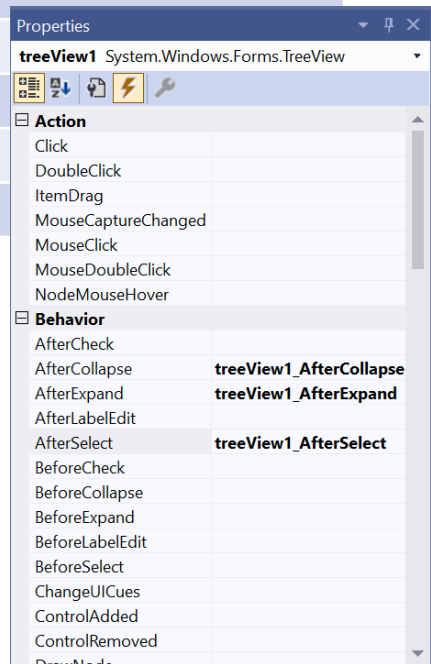


TreeView

주요 속성	설명
SelectedNode	선택된 노드 확인
CheckBoxes	노드 옆에 확인란 표시
ShowPlusMinus	노드를 확장하거나 축소할 수 있는 더하기 또는 빼기 기호 단추 표시 여부 설정
ShowRootLines	모든 루트 트리 노드를 연결하는 선 표시 여부 설정
ShowLines	자식 트리 노드를 해당 루트 노드에 연결하는 선 표시 여부 설정
HotTracking	마우스 포인터가 지나갈 때 트리 노드 레이블의 모양이 하이퍼링크 모양으로 변경되도록 설정

TreeView

주요 이벤트	설명
AfterCheck	트리노드 확인란이 선택된 후에 발생
AfterCollapse	트리노드가 축소된 후에 발생
AfterExpand	트리노드가 확장된 후에 발생
AfterLabelEdit	트리노드 레이블 텍스트가 편집된 후에 발생
AfterSelect	트리노드가 선택된 후에 발생



TreeNode

- 트리뷰를 구성하는 노드
- 트리뷰의 Nodes 컬렉션을 구성하는 요소

TreeNode 편집기를 사용하여 각 노드 생성 및 속성설정

The image shows two windows from Visual Studio. On the left is the 'Properties' window for a 'treeView1 System.Windows.Forms.TreeView'. The 'Nodes' property is highlighted with a red box and a red arrow pointing to the right. On the right is the 'TreeNode Editor' dialog. It has a tree view on the left with a tree structure: '지역명' (expanded) containing '강원도' (expanded) with sub-nodes '춘천', '강릉', '경기도' (expanded) with sub-nodes '과천', '남양주', '제주도' (expanded) with sub-nodes '제주시', '서귀포시'. On the right of the dialog is the 'Node0 properties' section with 'Appearance' and 'Behavior' tabs. The 'Appearance' tab is active, showing properties like BackColor, ForeColor, Name (Node0), NodeFont (none), Text (지역명), and ToolTipText. The 'Behavior' tab shows properties like Checked (False), ContextMenu (none), ContextMenuStrip (none), ImageIndex (default), ImageKey (default), Index (0), and SelectedImageIndex (default). At the bottom of the dialog are 'Add Boot', 'Add Child', 'OK', and 'Cancel' buttons.

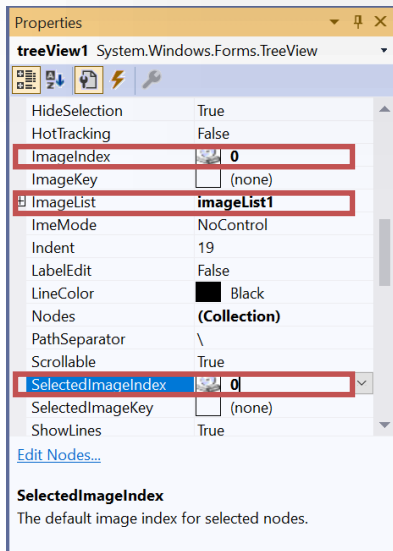
트리 뷰 탐색 속성

- FirstNode, LastNode, NextNode, PrevNode, Parent, FullPath

TreeView - Add Icons

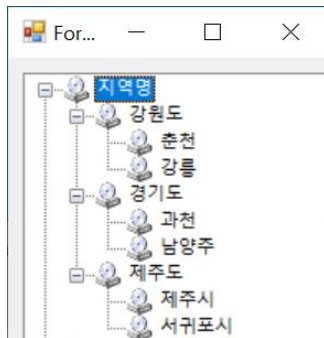
- 이미지리스트를 사용하여 각 노드의 아이콘이미지
- 설정의 **ImageIndex** 및 **SelectedImageIndex** 속성 이용

- ImageIndex : 보통 및 확장 상태의 노드에 대해 표시되는 이미지
- SelectedImageIndex : 선택된 상태의 노드에 대해 표시되는 이미지



//TreeView 의 Image 설정 코드

```
treeView1.ImageList = this.imageList1;  
treeView1.ImageIndex = 0;  
treeView1.SelectedImageIndex = 0;
```



TreeView

//TreeNode

```
System.Windows.Forms.TreeNode treeNode1 = new System.Windows.Forms.TreeNode("춘천");
System.Windows.Forms.TreeNode treeNode2 = new System.Windows.Forms.TreeNode("강릉");
System.Windows.Forms.TreeNode treeNode3 = new
    System.Windows.Forms.TreeNode("강원도",
        new System.Windows.Forms.TreeNode[] { treeNode1, treeNode2});
System.Windows.Forms.TreeNode treeNode4 = new System.Windows.Forms.TreeNode("과천");
System.Windows.Forms.TreeNode treeNode5 = new System.Windows.Forms.TreeNode("남양주");
System.Windows.Forms.TreeNode treeNode6 = new
    System.Windows.Forms.TreeNode("경기도",
        new System.Windows.Forms.TreeNode[] { treeNode4, treeNode5});
System.Windows.Forms.TreeNode treeNode7 = new System.Windows.Forms.TreeNode("제주시");
System.Windows.Forms.TreeNode treeNode8 = new System.Windows.Forms.TreeNode("서귀포");
System.Windows.Forms.TreeNode treeNode9 = new
    System.Windows.Forms.TreeNode("제주도",
        new System.Windows.Forms.TreeNode[] { treeNode7, treeNode8});
System.Windows.Forms.TreeNode treeNode10 = new
    System.Windows.Forms.TreeNode("지역명",
        new System.Windows.Forms.TreeNode[] { treeNode3, treeNode6, treeNode9});
```

//TreeView

```
System.Windows.Forms.TreeView treeView1 = new System.Windows.Forms.TreeView();
```

//TreeView의 Nodes 속성을 통한 자식노드 추가

```
treeView1.Nodes.AddRange(new System.Windows.Forms.TreeNode[] { treeNode10 });
```

TreeView

//TreeView 메서드, 이벤트, 속성 설정 예

```
private void Form1_Load(object sender, EventArgs e) {  
    treeView1.ExpandAll(); //폼로딩 시 트리뷰의 모든 노드를 확장  
}
```

//트리 노드 확장 후 발생하는 이벤트 핸들러

```
private void treeView1_AfterExpand(object sender, TreeViewEventArgs e) {  
    MessageBox.Show("확장되었습니다.", "AfterExpand");  
}
```

//트리노드 선택 후 발생하는 이벤트 핸들러

```
private void treeView1_AfterSelect(object sender, TreeViewEventArgs e) {  
    textBox1.Text = e.Node.Text; //매개변수를 이용한 현재 선택된 노드 얻기  
    // textBox1.Text = treeView1.SelectedNode.Text; //선택된 노드를 얻기 위한 속성이용  
    if (e.Node.Parent != null)  
        textBox1.Text = e.Node.Parent.Text;  
    textBox1.Text = e.Node.FullPath; //현재 노드까지 루트로부터 전체 경로 얻기  
}
```

//버튼을 누르면 textBox2에 적혀있는 텍스트를 선택된 노드의 하위 노드에 추가

```
private void button1_Click(object sender, EventArgs e) {  
    treeView1.SelectedNode.Nodes.Add(textBox2.Text);  
}  
  
private void button2_Click(object sender, EventArgs e) {  
    TreeNode currNode = treeView1.SelectedNode; //현재 선택 노드  
    if (currNode.PrevNode != null) {  
        treeView1.SelectedNode = currNode.PrevNode; //현재 선택 노드의 이전 노드  
        treeView1.Focus();  
    }  
    else MessageBox.Show("이전노드는 존재하지 않습니다", "PreNode");  
}
```

