



Windows Controls

Giảng viên: BÙI NGỌC LÊ

Nội Dung

- **Các control thông dụng**
 - ❑ **PictureBox, ImageList**
 - ❑ **TrackBar**
 - ❑ **NumericUpDown**
 - ❑ **DomainUpDown**
 - ❑ **ProgressBar**

PictureBox

- Sử dụng để hiển thị ảnh dạng bitmap, metafile, icon, JPEG, GIF.
- Sử dụng thuộc tính Image để thiết lập ảnh lúc design hoặc runtime.
- Các thuộc tính
 - Image: ảnh cần hiển thị
 - SizeMode:
 - Normal
 - StretchImage
 - AutoSize
 - CenterImage
 - Zoom

PictureBox



**5 pictureBox với
các SizeMode
tương ứng**

ImageList

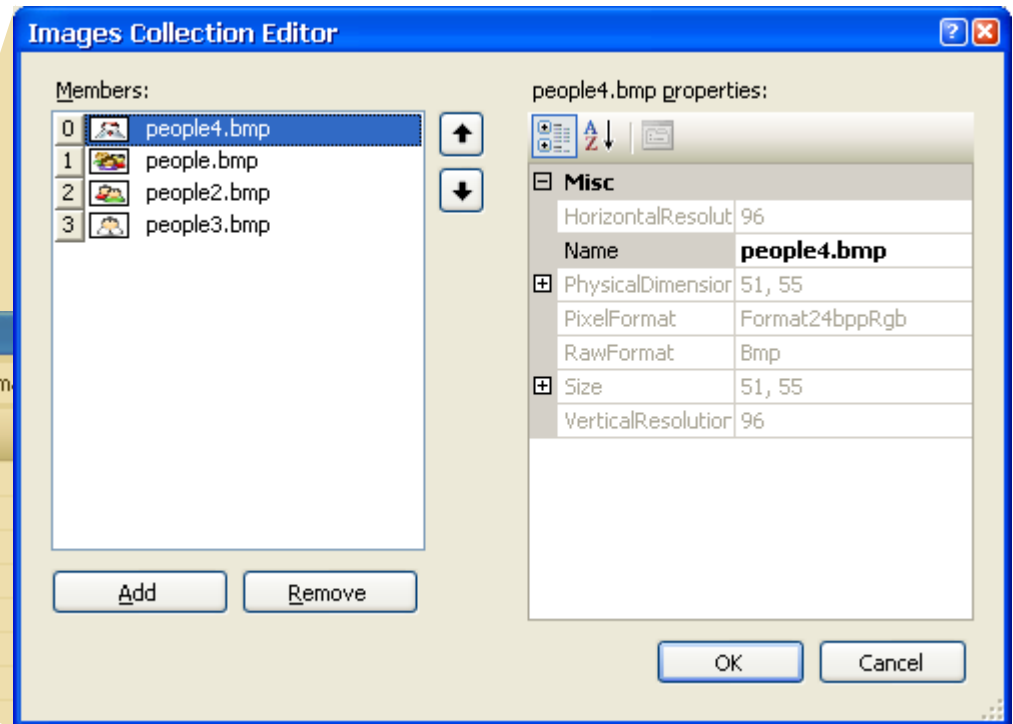
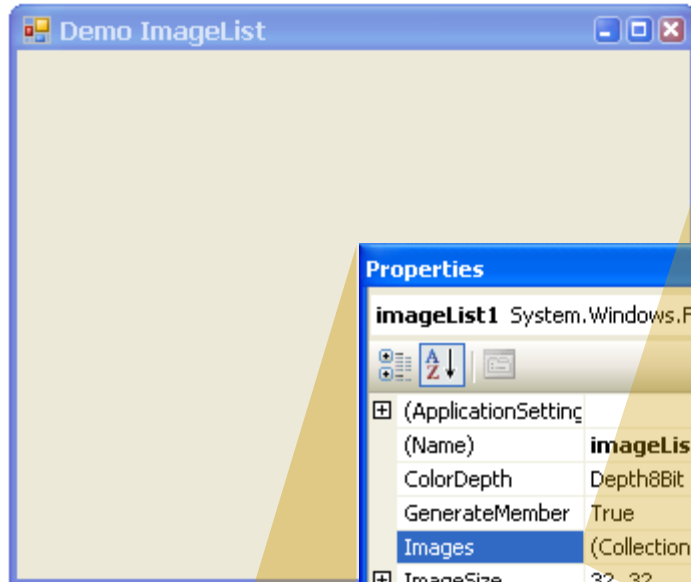
- **Cung cấp tập hợp những đối tượng image cho các control khác sử dụng**
 - **ListView**
 - **TreeView**
- **Các thuộc tính thường dùng**
 - **ColorDepth**: độ sâu của màu
 - **Images**: trả về ImageList.ImageCollection
 - **ImageSize**: kích thước ảnh
 - **TransparentColor**: xác định màu sẽ transparent

ImageList

- **Các bước sử dụng ImageList**
 - Kéo control ImageList từ ToolBox thả vào Form
 - Thiết lập kích thước của các ảnh: ImageSize
 - Bổ sung các ảnh vào ImageList qua thuộc tính Images
 - Sử dụng ImageList cho các control
 - Khai báo nguồn image là image list vừa tạo cho control
 - Thường là thuộc tính ImageList
 - Thiết lập các item/node với các ImageIndex tương ứng
 - Việc thiết lập có thể ở màn hình design view hoặc code view

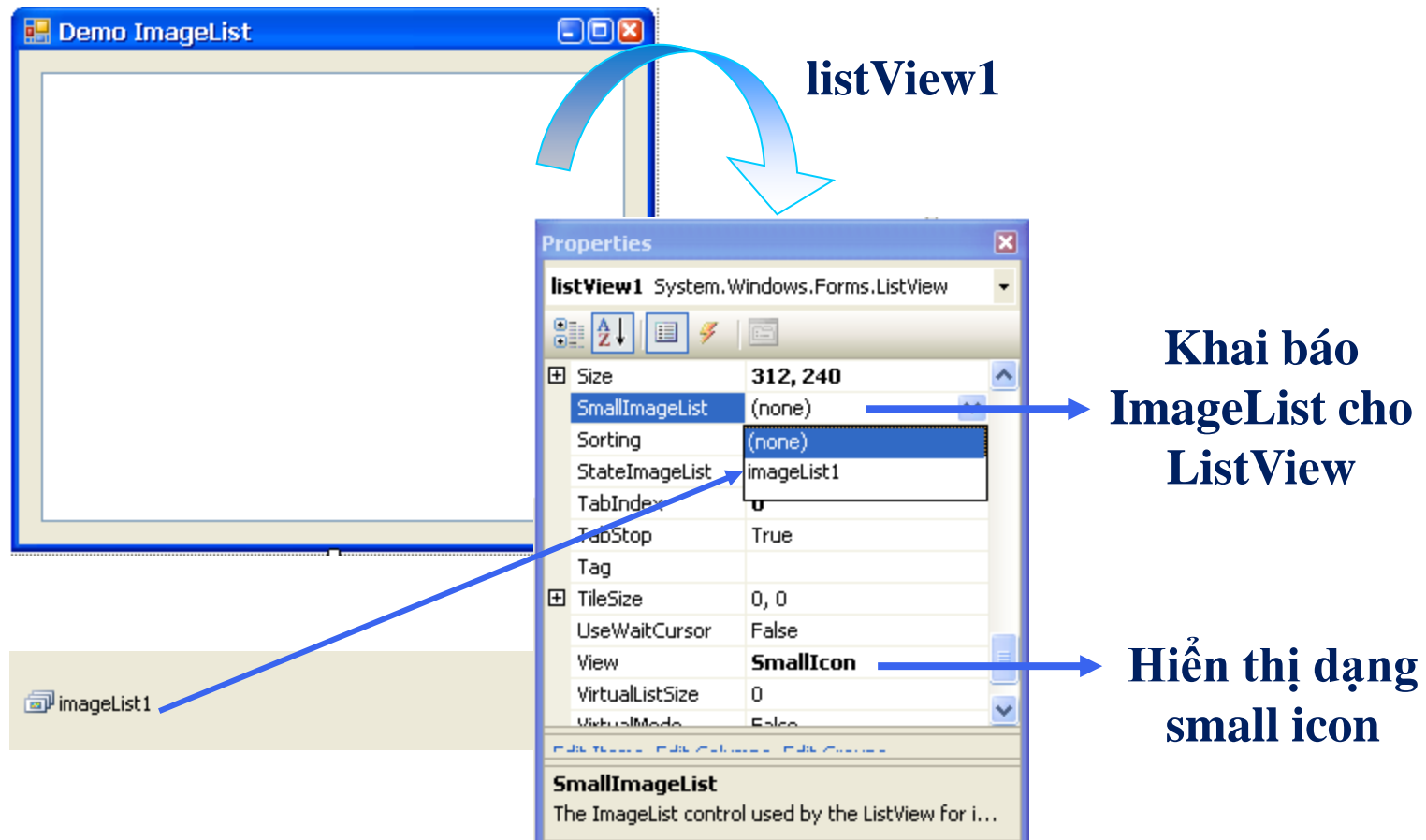
ImageList

■ Tạo ImageList



ImageList

■ Sử dụng ImageList trong ListView



The screenshot shows a Windows Forms application titled "Demo ImageList". It contains a `listView1` control. A blue arrow points from the `listView1` text to the `listView1` control in the Properties window. The Properties window shows the following properties:

Property	Value
Size	312, 240
SmallImageList	(none)
Sorting	(none)
StateImageList	imageList1
TabIndex	0
TabStop	True
Tag	
TileSize	0, 0
UseWaitCursor	False
View	SmallIcon
VirtualListSize	0
VirtualMode	False

Annotations:

- A blue arrow points from the `imageList1` control in the component tray to the `StateImageList` property in the Properties window.
- A blue arrow points from the `SmallImageList` property in the Properties window to the text "Khai báo ImageList cho ListView".
- A blue arrow points from the `View` property in the Properties window to the text "Hiển thị dạng small icon".

ImageList

■ Thêm Item

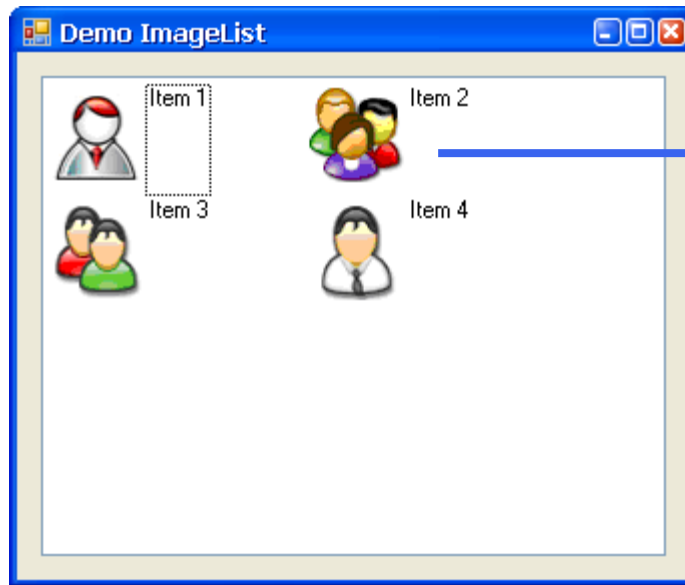
The screenshot illustrates the steps to add an item to a `ListView` in Visual Studio:

- The **Properties** window for `listView1` (System.Windows.Forms.ListView) shows the **Items** property selected, which is a **(Collection)**.
- The **ListViewItem Collection Editor** window shows the collection of items. The list contains four items: `ListItem: {Item 1}`, `ListItem: {Item 2}`, `ListItem: {Item 3}`, and `ListItem: {Item 4}`. The `Item 4` is currently selected.
- The **Appearance** tab for the selected `ListItem: {Item 4}` is open. It shows various properties for the item's appearance and behavior. The **ImageIndex** property is highlighted, showing a value of **3**.

A blue arrow points from the **ImageIndex** value (3) to the text: **Khái báo image cho item qua ImageIndex**.

ImageList

■ Demo



**Mỗi item sẽ có ảnh
theo đúng thứ tự
ImageIndex được
khai báo trong
ImageList**

TrackBar

- Cho phép user thiết lập giá trị trong khoảng cố định cho trước
- Thao tác qua thiết bị chuột hoặc bàn phím

Properties

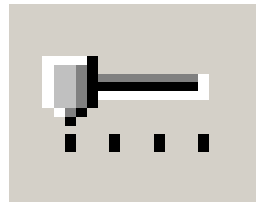
Minimum

Maximum

TickFrequency

TickStyle

Value



ValueChanged
Scroll

Methods

SetRange

TrackBar

```
public void AddTrackBar() {
```

```
    TrackBar tb1 = new TrackBar();  
    tb1.Location = new Point(10, 10);  
    tb1.Size = new Size(250, 50);
```

Tạo thể hiện

```
    tb1.Minimum = 0;  
    tb1.Maximum = 100;
```

Thiết lập khoảng: 0 - 100

```
    tb1.SmallChange = 1;  
    tb1.LargeChange = 5;
```

Số vị trí di chuyển khi dùng
phím mũi tên

```
    tb1.TickStyle = TickStyle.BottomRight;
```

Số vị trí di chuyển
khi dùng phím Page

```
    tb1.TickFrequency = 10;  
    tb1.Value = 10;  
    Controls.Add(tb1);
```

Số khoảng cách giữa
các tick mark

Kiểu stick ở bên
dưới/bên phải track

TrackBar

- Bổ sung Label hiển thị giá trị của TrackBar

The screenshot illustrates the process of adding a Label to display the value of a TrackBar. The main window, titled 'Form1', contains a TrackBar control and a Label control named 'label1'. The 'Properties' window for 'trackBar1' is open, showing the 'Scroll' event selected. The event handler code is displayed in a text box:

```
private void trackBar1_Scroll(object sender, EventArgs e)
{
    // label1 hiển thị giá trị của trackbar
    label1.Text = trackBar1.Value.ToString();
}
```

NumericUpDown

- Cho phép user chọn các giá trị trong khoảng xác định thông qua
 - Nút up & down
 - Nhập trực tiếp giá trị
- Các thuộc tính
 - Minimum
 - Maximum
 - Value
 - Increment
- Sự kiện
 - ValueChanged
- Phương thức
 - DownButton
 - UpButton

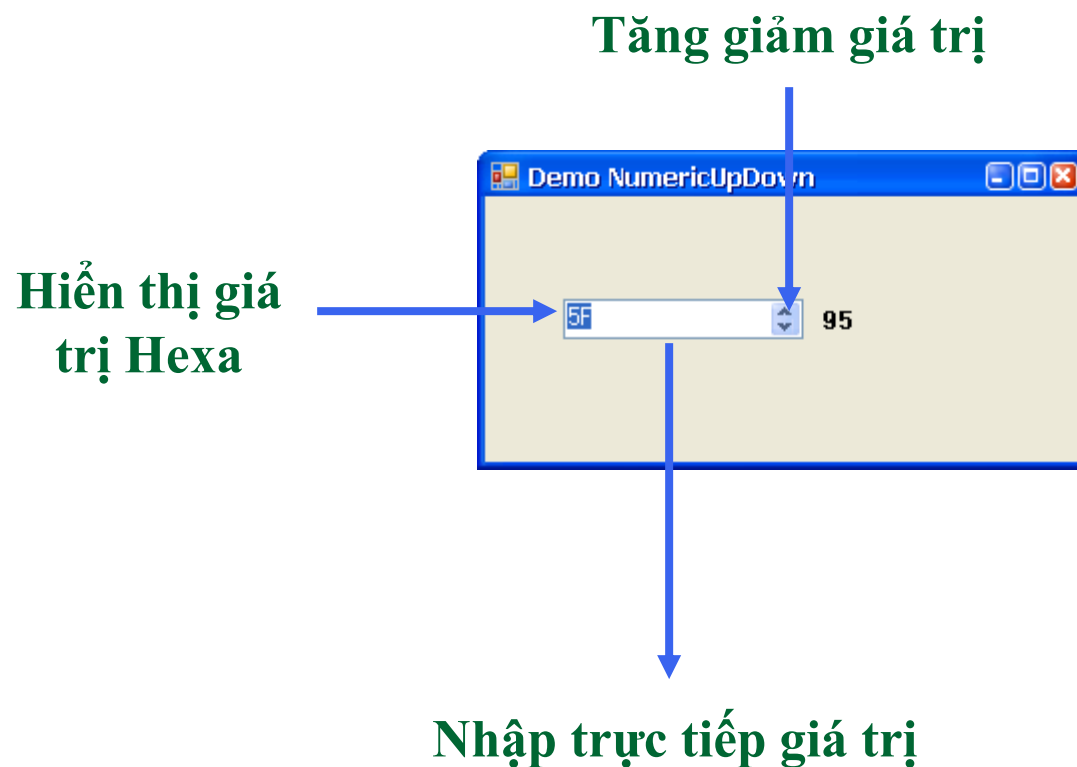
NumericUpDown

■ Đoạn code thêm control NumericUpDown

```
public void AddNumericUpDown() {  
    NumericUpDown numUpDn = new NumericUpDown();  
    numUpDn.Location = new Point(50, 50);  
    numUpDn.Size = new Size(100, 25);  
  
    numUpDn.Hexadecimal = true;           // hiển thị dạng hexa  
    numUpDn.Minimum = 0;                  // giá trị nhỏ nhất  
    numUpDn.Maximum = 255;                // giá trị lớn nhất  
    numUpDn.Value = 0xFF;                 // giá trị khởi tạo  
    numUpDn.Increment = 1;                // bước tăng/giảm  
  
    Controls.Add(numUpDn);                // thêm control vào ds control của form  
}
```

NumericUpDown

■ Demo

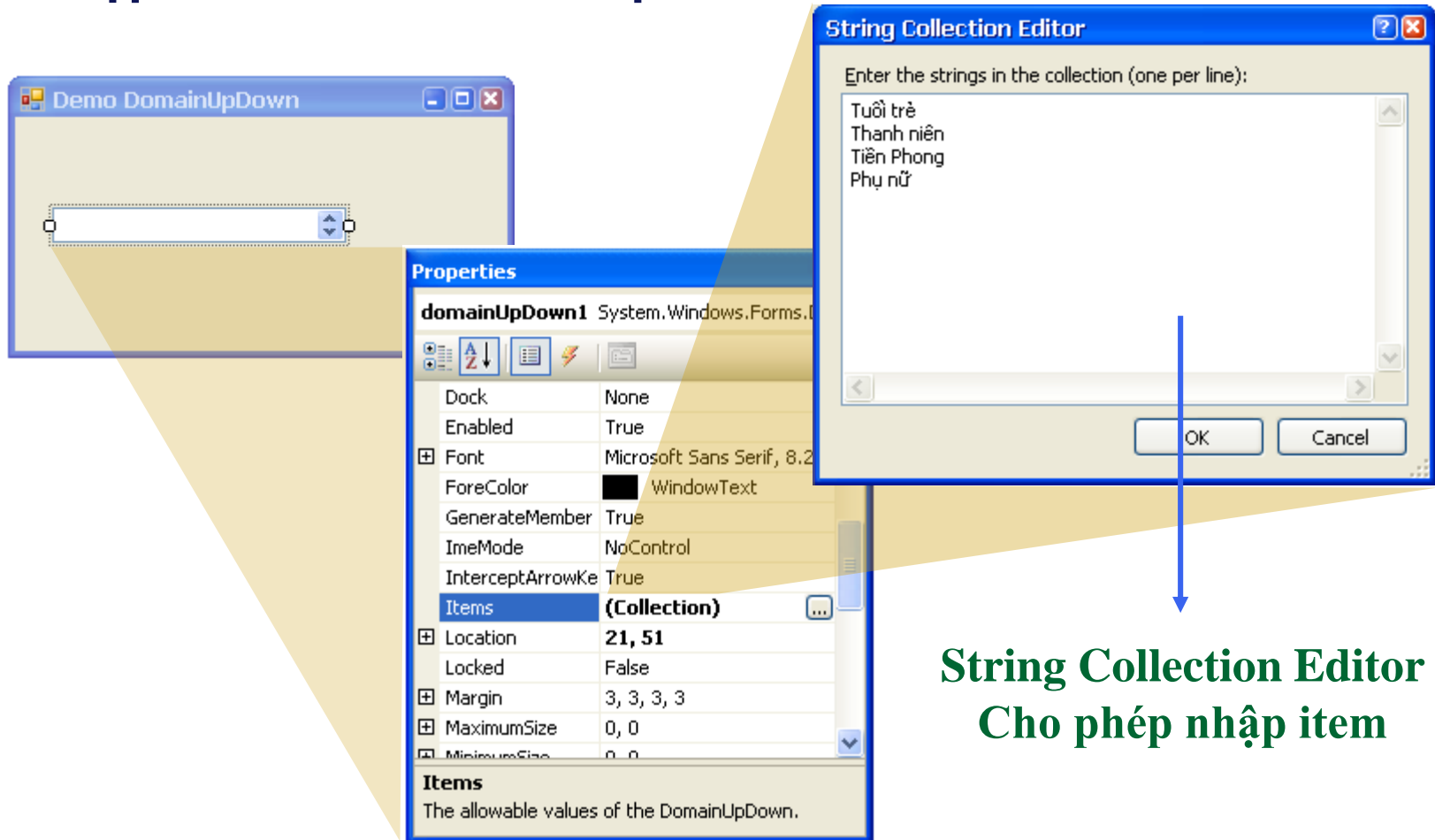


DomainUpDown

- Cho phép user chọn item trong sổ danh sách item thông qua
 - Button Up & Down
 - Nhập từ bàn phím
- Properties
 - Items: danh sách item
 - ReadOnly: true chỉ cho phép thay đổi giá trị qua Up & Down
 - SelectedIndex: chỉ mục của item đang chọn
 - SelectedItem: item đang được chọn
 - Sorted: sắp danh sách item
 - Text: text đang hiển thị trên DomainUpDown.
- Event
 - SelectedItemChanged

DomainUpDown

■ Nhập item cho DomainUpDown



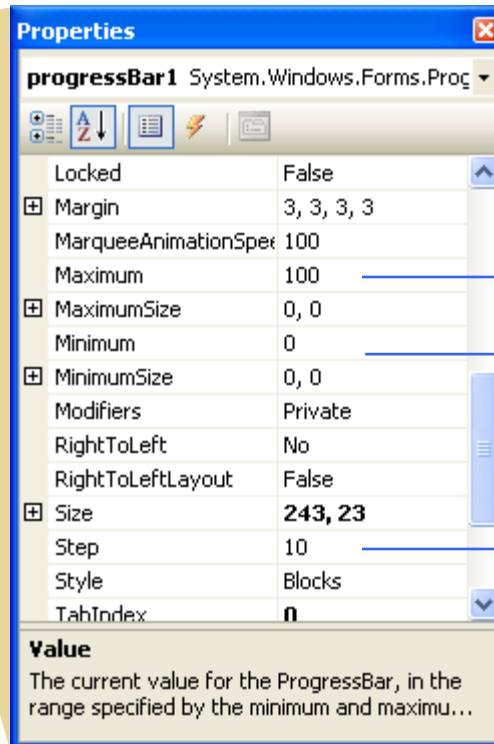
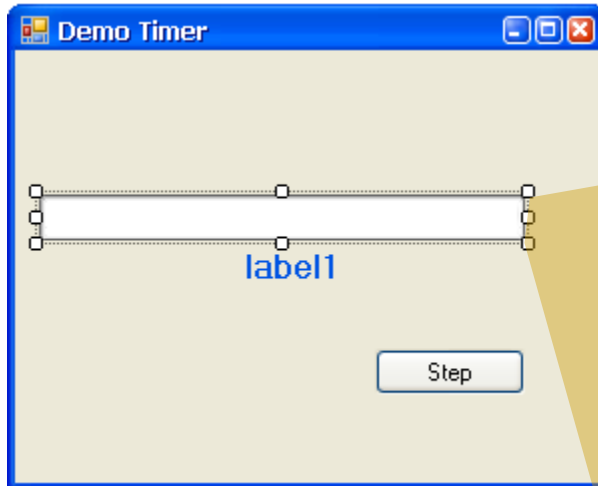
String Collection Editor
Cho phép nhập item

ProgressBar

- **Hiển thị tiến độ thực hiện của một công việc nào đó**
- **Các thuộc tính**
 - **Minimum:** giá trị nhỏ nhất
 - **Maximum:** giá trị lớn nhất
 - **Step:** số bước tăng khi gọi hàm PerformStep
 - **Value:** giá trị hiện tại
 - **Style:** kiểu của progress bar
- **Phương thức**
 - **PerformStep():** tăng thêm step
 - **Increment(int value):** tăng vị trí hiện tại của tiến độ với giá trị xác định

ProgressBar

- Khai báo thanh tiến độ 0-100, step = 10

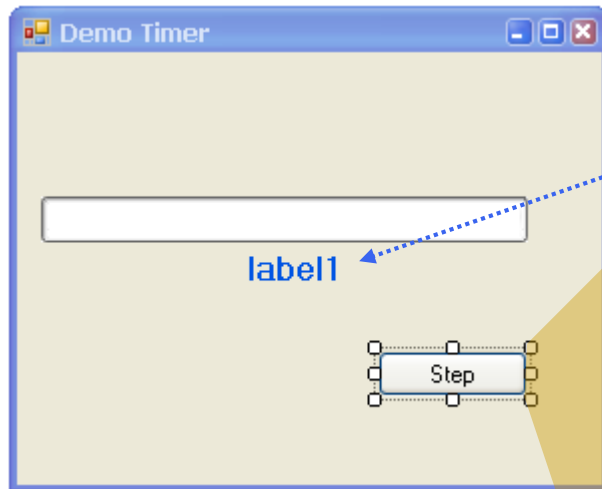


Max = 100

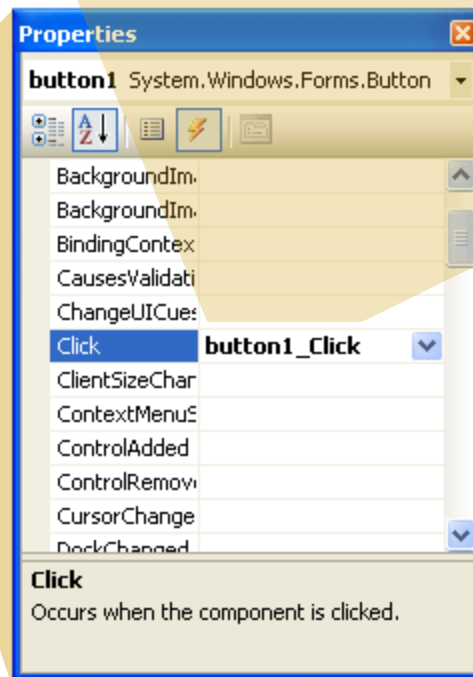
Min = 0

Step = 10

ProgressBar

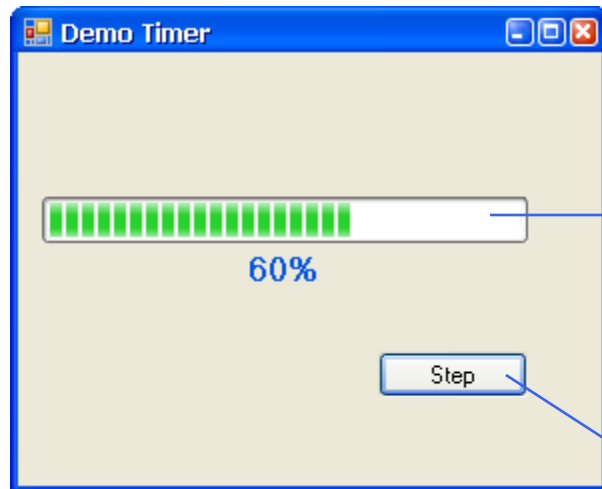


```
private void button1_Click(object sender, EventArgs e)
{
    // tăng thêm một bước: value += step
    progressBar1.PerformStep();
    // hiển thị tiến độ % lên label
    label1.Text = progressBar1.Value.ToString() + "%";
}
```



ProgressBar

■ Demo



Thể hiện trực
quan tiến độ

Tăng tiến độ theo
step và cập nhật lại
% hoàn thành lên
label