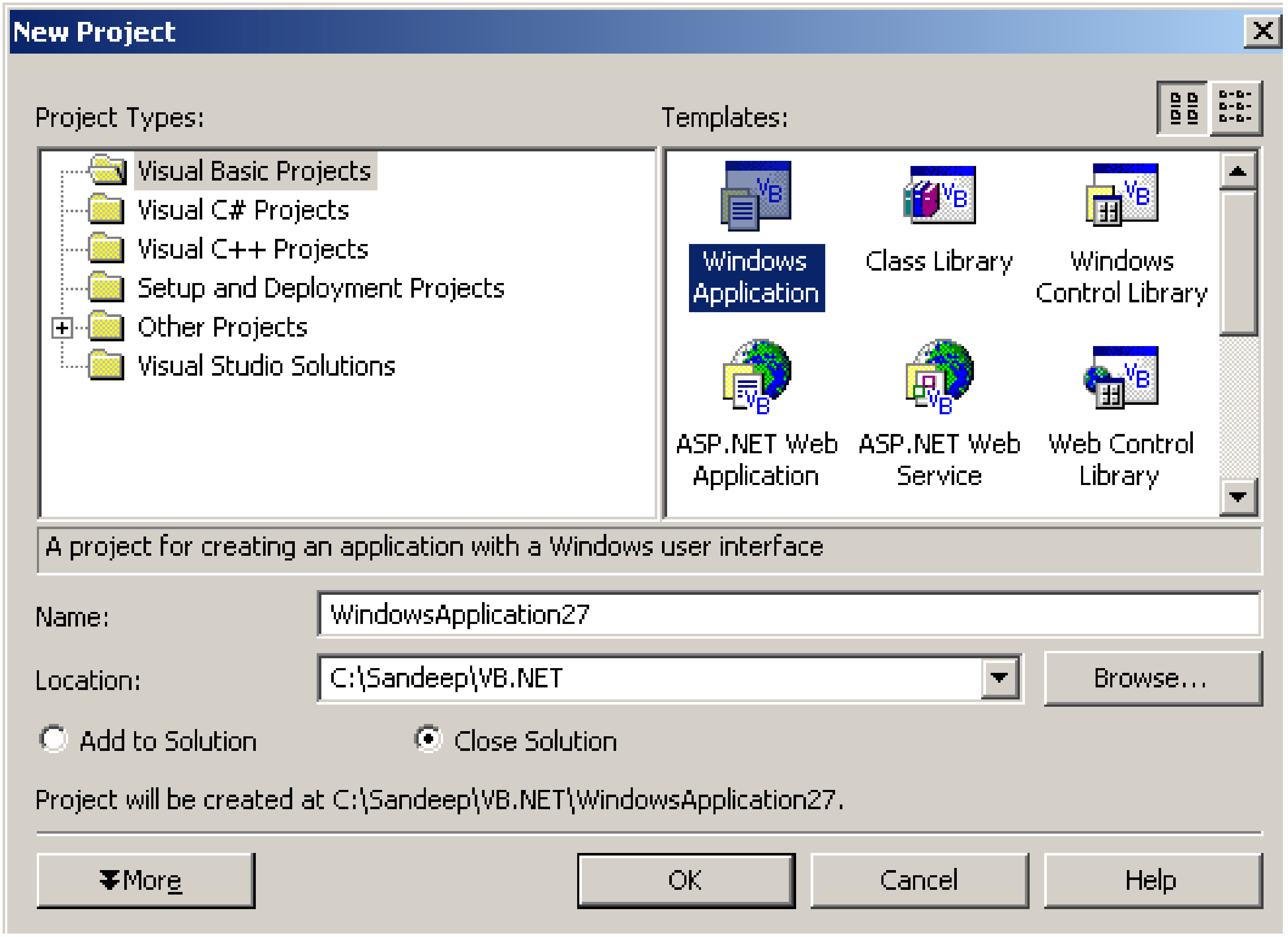
VISUAL BASIC CONTROLS

Windows Forms

- They are the base on which we build, develop all our user interface and they come with a rich set of classes.
- Forms allow us to work visually with controls and other items from the toolbox.
- In VB .NET forms are based on the System.Windows.Forms namespace and the form class is System.Windows.Forms.Form.
- The form class is based on the Control class which allows it to share many properties and methods with other controls.

- VB.Net and .Net provide extensive support for building Windows Applications.
- The most important point about windows applications is that they are 'event driven'.
- All windows applications present a graphical interface to their users and respond to user interaction.
- This graphical user interface is called a 'Windows Form' or a 'WinForm' for short.
- A windows form may contain text labels, push buttons, text boxes, list boxes, images, menus and a vast range of other controls.
- In fact, a WinForm is also a windows control just like a text box, label, etc.

- When we open a new project in Visual Basic the dialogue box that appears first is the one which looks like the image below.



New Project

Project Types:

- Visual Basic Projects
- Visual C# Projects
- Visual C++ Projects
- Setup and Deployment Projects
- Other Projects
- Visual Studio Solutions

Templates:

- Windows Application
- Class Library
- Windows Control Library
- ASP.NET Web Application
- ASP.NET Web Service
- Web Control Library

A project for creating an application with a Windows user interface

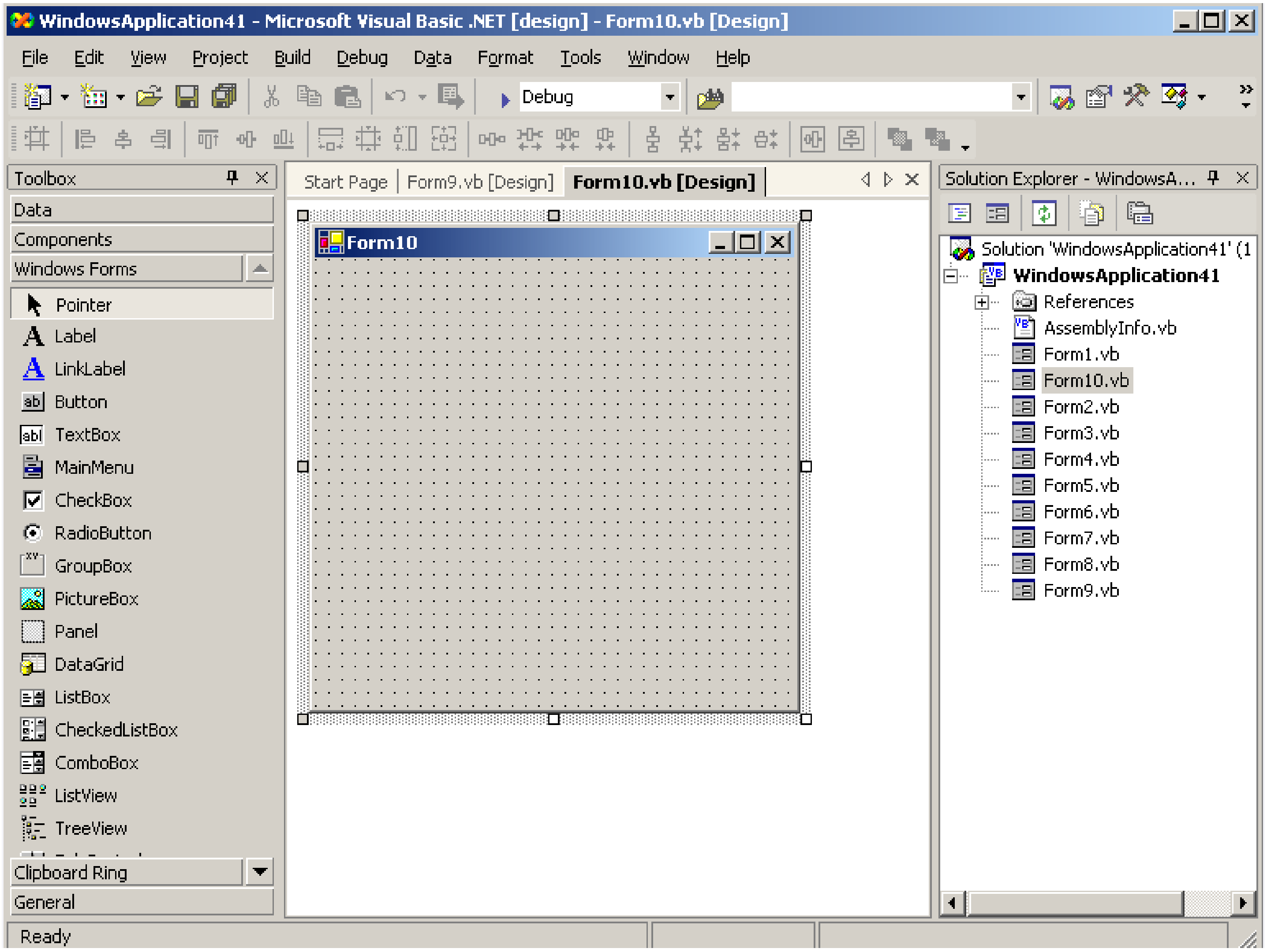
Name: WindowsApplication27

Location: C:\Sandeep\VB.NET

Add to Solution [✓ Close Solution]

Project will be created at C:\Sandeep\VB.NET\WindowsApplication27.

- When we open application window in Visual Basic the dialogue box that appears first is the one which looks like the image below.



Controls

- A control is an object that can be drawn on to the Form to enable or enhance user interaction with the application.
- Examples of these controls, TextBoxes, Buttons, Labels, Radio Buttons, etc.
- All these Windows Controls are based on the Control class, the base class for all controls.
- VB.NET allows us to work with controls in two ways: at design time and at runtime.

- Working with controls at design time means, controls are visible to us and we can work with them by dragging and dropping them from the Toolbox and setting their properties in the properties window.
- Working at runtime means, controls are not visible while designing, are created and assigned properties in code and are visible only when the application is executed.
- The Control class is in the System.Windows.Forms namespace.
- It is a base class for the Windows Controls.

Common properties

- **<u>Name</u>:** This property uniquely Identify control/object on form.
- **<u>Enabled:</u>** Return /sets a value that determines whether an object can respond to user-generated events like click, double click etc.
  - Its value is either true or false.
  - Default value: "true"
  - If value is set to false, then object will not respond user-generated events.

- **Visible:** Return/sets a value that determines whether object is visible or hidden.
  - Its value is either true or false.
  - Default value: True.
  - Set to false to hide object.
- **Backcolor:** Return/set background color used to display teat and graphics in object.
- **Forecolor:** Return/sets foreground color used to display text and graphic in an object.
- **Font:** Return font object.

- **Size:** Return/sets the height and width of object.

- **Left:** Return/sets the distance between internal left edge of object and left edge of its container.
- **TabIndex:** Return/sets tab order of object within its parent form.
- **TabStop:** Default: true.

  Return/sets a value indicating whether user can use “TAB” key to give focus to an object.
- **Index**: return/set number identifying in control array.
- **Top**: Returns/sets distance between internal top edge of object and Top edge of its container.

Label

- Labels are those controls that are used to display text in other parts of the application.
- Notable property of the label control is the <u>text</u> property which is used to set the text for the label.

TextBox Control

- TextBoxes are used to accept input from the user or used to display text.
- By default we can enter up to 2048 characters in a TextBox but if the Multiline property is set to True we can enter up to 32KB of text.
- **Some Notable Properties:**
- <u>Enabled:</u> Default value is True. To disable, set the property to False.
- <u>Multiline:</u> Setting this property to True makes the TextBox multiline which allows to accept multiple lines of text. Default value is False.

- PasswordChar: Used to set the password character. The text displayed in the TextBox will be the character set by the user. Say, if you enter *, the text that is entered in the TextBox is displayed as *.
- ReadOnly: Makes this TextBox readonly. It doesn't allow to enter any text.
- TextAlign: Allows to align the text from three possible options. The default value is left and you can set the alignment of text to right or center.
- Scrollbars: Allows to add a scrollbar to a Textbox. Very useful when the TextBox is multiline. You have four options with this property. Options are None, Horizontal, Vertical and Both.

Button Control

- They are the controls which we click and release to perform some action.
- Buttons are used mostly for handling events in code, say, for sending data entered in the form to the database and so on.
- Default event: Click

-
  ```
  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As_
  System.EventArgs) Handles Button1.Click
              'You place the code here to perform action when Button is clicked
  End Sub
  ```

- With the help of BackColor and BackgroundImage properties we can set a background color and a background image to the button.
- We change the appearance style of the button with the FlatStyle property.
- with the TextAlign property we can set where on the button the text should appear from a predefined set of options.
- With the Location property you can change the location of the button.

CheckBox

- CheckBoxes are those controls which gives us an option to select, say, Yes/No or True/False.
- When a checkbox is selected a check (a tick mark) appears indicating a selection.
- The CheckBox control is based on the TextBoxBase class which is based on the Control class.
- <u>CheckAlign:</u> Used to set the alignment for the CheckBox from a predefined list.
- <u>Checked:</u> Default value is False, set it to True if you want the CheckBox to be displayed as checked.

- <u>CheckState:</u> Default value is Unchecked. Set it to True if you want a check to appear. When set to Indeterminate it displays a check in gray background.
- <u>FlatStyle:</u> Default value is Standard. Select the value from a predefined list to set the style of the checkbox.

RadioButton

- Like CheckBoxes, RadioButtons are used to select and deselect options but they allow us to choose from mutually exclusive options.
- The RadioButton control is based on the ButtonBase class which is based on the Control class.
- A major difference between CheckBoxes and RadioButtons is, RadioButtons are mostly used together in a group.

- <u>CheckAlign:</u> Used to set the alignment for the RadioButton from a predefined list.
- <u>Checked:</u> Default value is False, set it to True if you want the RadioButton to be displayed as checked.
- <u>FlatStyle:</u> Default value is Standard. Select the value from a predefined list to set the style of the RadioButton.

Date time picker

- It is used to allow the user to select a date and time, and to display that date/time in the specified format. We can limit the dates and times that can be selected by setting the MinDate and MaxDate properties.
- **CustomFormat** A string that represents the custom date/time format. The default is a null reference.
- **ShowUpDown** if it is set to true, an up-down control is used to adjust time values instead of the drop-down calendar. The date/time can be adjusted by selecting each element individually and using the up and down buttons to change the value
- **Format** – It gives the following format

Member name	Description
Custom	The DateTimePicker control displays the date/time value in a custom format.
Long	The DateTimePicker control displays the date/time value in the long date format set by the user's operating system.
Short	The DateTimePicker control displays the date/time value in the short date format set by the user's operating system.
Time	The DateTimePicker control displays the date/time value in the time format set by the user's operating system.

ListBox and combobox

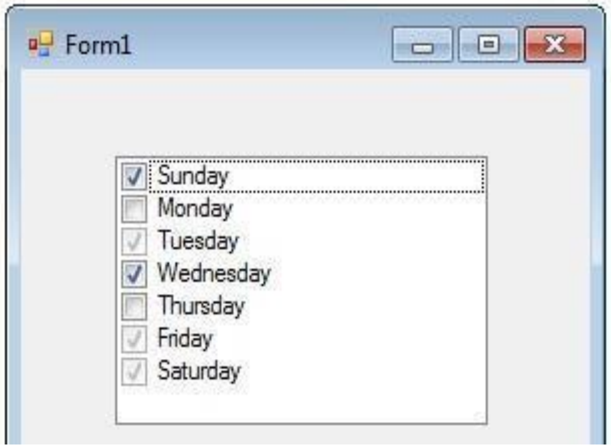
- The **ListBox** and **ComboBox** control helps to display a list of items to the user that the user can select by clicking. Combobox allows only one to get selected while listbox allows multiple selections.
- **SelectionMode** property selects single or multiple items from the ListBox.
- **MultiColumn** property displays items in multiple columns instead of a straight vertical list of items. This allows the control to display more visible items and prevents the need for the user to scroll to an item.
- **Sorted** property is used to sort the data's displayed in the listbox.
- **Items Collection-** It can be used to support the following methods.
  - **Add** method provides the ability to add a single object to the collection.
  - **Remove** method can be used to remove the specified data from the ListBox.
  - **RemoveAt** method can be used to remove data at the specified position from the ListBox.
  - **Clear** method to remove all the items from the list.
  - **Count** returns the total no. of elements in the list.
- **MaxDropDownItems**-The maximum number of items of in the drop-down portion. The minimum for this property is 1 and the maximum is 100.

DropDownStyle property supported only by dropdown list. It has the following constants.

Member name	Description
DropDown	The text portion is editable. The user must click the arrow button to display the list portion.
DropDownList	The user cannot directly edit the text portion. The user must click the arrow button to display the list portion.
Simple	The text portion is editable. The list portion is always visible.

CheckedListBox:

The **CheckedListBox** control gives you all the capability of a list box and also allows you to display a check mark next to the items in the list box.



Properties	Description
HorizontalScrollbar	Property used to Get or set if a checked list box should display a horizontal scroll bar.
Item Height	Property returns the height of an item.
Items	Property returns the items of this control.
MultiColumn	Property to get or set the if the checked list box allows multicolumn.
SelectedIndex	Property used to set or get the selected item in the control.
SelectedItem	Property used to get or set the selected item.
SelectedItems	Property used to get or set the selected items.
Locked	Prevents the control being moved at design time.

picturebox

- **PictureBox Control** is used to display graphics from a bitmap, metafile, icon, JPEG, GIF or PNG file. Set the Image property to the Image you want to display, either at design time or at run time. Clipping and positioning of the image in the display area is controlled by the SizeMode property which is set to values in the PictureBoxSizeMode enumeration. The PictureBox control is displayed by default without any borders. To provide a standard or three-dimensional border using the BorderStyle property to distinguish the picture box from the rest of the form, even if it contains no image. The SizeMode can be any of the below values.

Member name	Description
AutoSize	The PictureBox is sized equal to the size of the image that it contains.
CenterImage	The image is displayed in the center if the PictureBox is larger than the image. If the image is larger than the PictureBox, the picture is placed in the center of the PictureBox and the outside edges are clipped.
Normal	The image is placed in the upper-left corner of the PictureBox. The image is clipped if it is larger than the PictureBox it is contained in.
StretchImage	The image within the PictureBox is stretched or shrunk to fit the size of the PictureBox.

RichTextBox

- The **RichTextBox** control allows the user to enter and edit text. It also provides more advanced formatting features than the standard TextBox control. It provides the following features.
- Text can be assigned directly to the control, or can be loaded from a Rich Text Format (RTF) or plain text file.
- The text within the control can be assigned character and paragraph formatting.
- The **RichTextBox** control provides a number of properties you can use to apply formatting to any portion of text within the control. To change the formatting of text, it must first be selected. Only selected text can be assigned character and paragraph formatting. Once a setting has been made to a selected section of text, all text entered after the selection is also formatted with the same settings until a setting change is made or a different section of the control's document is selected.

- SelectionFont property changes the text to bold or italic. This property can be used to change the size and Font type of the text.
- SelectionColor property changes the color of the text.
- SelectionBullet property can be used to create bulleted lists.
- SelectionIndent, SelectionRightIndent, and SelectionHangingIndent properties can be used to adjust paragraph formatting.
- LoadFile method can be used to load an existing RTF or ASCII text file into the control. SaveFile method can be used to save a file to RTF or ASCII text.

- The **RichTextBox** is typically used to provide text manipulation and display features similar to word processing applications such as Microsoft Word.

Treeview

- The TreeView control is used to display a hierarchy of nodes (both parent, child).
- You can expand and collpase these nodes by clicking them.
- This control is similar to Windows Explorer which displays a tree view in it's left pane to list all the folders on the hard disk.

Notable Properties of TreeView

- Bounds: Gets the actual bound of the tree node
- Checked: Gets/Sets whether the tree node is checked
- FirstNode: Gets the first child tree node
- FullPath: Gets the path from the root node to the current node
- ImageIndex: Gets/Sets the image list index of the image displayed for a node
- Index: Gets the location of the node in the node collection
- IsEditing: Gets whether the node can be edited

- IsExpaned: Gets whether the node is expaned
- IsSelected: Gets whether the node is selected
- LastNode: Gets the last child node
- NextNode: Gets the next sibling node
- NextVisibleNode: Gets the next visible node
- NodeFont: Gets/Sets the font for nodes
- Nodes: Gets the collection of nodes in the current node
- Parent: Gets the parent node of the current node
- PrevNode: Gets the previous sibling node
- PrevVisibleNode: Gets the previous visible node

ToolTip Control

- This class allows providing help to users when they place the mouse cursor over a control. The ToolTip class is typically used to alert users to the intended use of a control. For example, to specify ToolTip text for a TextBox control that accepts a name, specifying the format of the name to typed into the control. In order for ToolTip text to be displayed when the user moves the mouse cursor over a control, the ToolTip text to be displayed must be associated with the control within an instance of the ToolTip class. To associate ToolTip text with a control, use the SetToolTip method.
- **AutomaticDelay** property enables to set a single delay value which is then used to set the values of the AutoPopDelay, InitialDelay, and ReshowDelay properties. Each time the AutomaticDelay property is set, the following values are set by default.

Property	Default Value
AutoPopDelay	10 times the AutomaticDelay property value.
InitialDelay	Equal to the AutomaticDelay property value.
ReshowDelay	1/5 of the AutomaticDelay property value.

- AutopopDelay -The period of time (in milliseconds) that the ToolTip remains visible when the mouse pointer is stationary within a control. The default value is 5000.
- InitialDelay - The period of time (in milliseconds) that the mouse pointer must remain stationary within a control before the ToolTip window is displayed.
- ReshowDelay - The length of time that must transpire before subsequent ToolTip windows appear as the mouse pointer moves from one control to another.

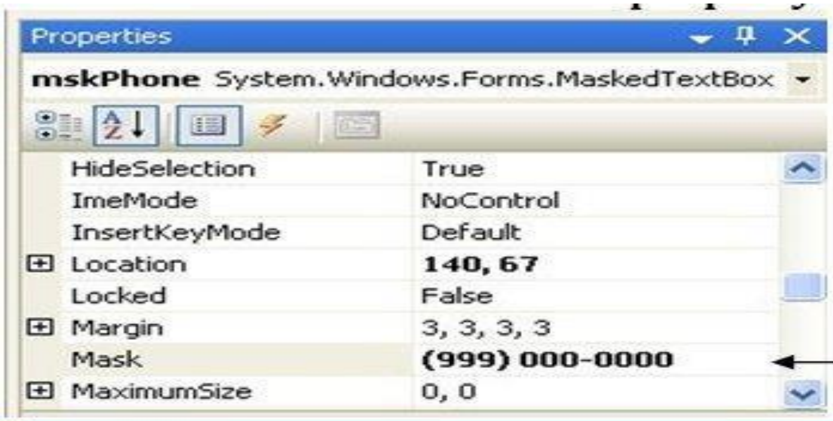
Progressbar

- A ProgressBar control visually indicates the progress of a lengthy operation. The ProgressBar control displays a bar that fills in from left to right with the system highlight color as an operation progresses. The ProgressBar control is typically used when an application performs tasks such as copying files or printing documents. Users of an application might consider an application unresponsive if there is no visual cue. By using the ProgressBar in an application, we can alert the user that the application is performing a lengthy task and that the application is still responding.
- **Minimum** specifies the minimum value with which the progressbar will start.
- **Maximum** specifies the maximum value with which the progressbar will terminate.
- **Value** the current value of the progressbar
- **Step** the value by which every time the bar in progressbar should increment.

Masked Textbox

- MaskedTextBox control is a specialized form of the Textbox control. You can specify the format (the Mask property) of the data required of the user. For example, you can select a mask for a ZIP code, a date, a phone number, or a social security number.
- Masked TextBox Control in .NET is used to restrict the input from user .
- A MaskedTextBox control provides validation mechanism for user input on a Form. For example, if you want a TextBox to accept date in mm/dd/yyyy format, you can set masking in the MaskedTextBox.
- MaskedTextBox accepts text input of a specific format. We often require phone numbers to have their area code and also the correct number of digits. To solve this problem, we use the MaskedTextBox control in Windows Forms.

Properties	Description
TextAlign	Text alignment is set using this property
Mask	This property is used to set the predefined mask description.
TextMaskFormat	This property is used to gets or sets a value that determines the literals, prompt character to be included in the formatted string.
HidePromptOnLeave	Property used to hide the mask characters, literals when the focus is lost from the control.
AllowPromptAsInput	Property used to set the input character as per the specified by moving character.
TextMaskFormat	Property used to specify the formatting for the text.
RejectInputOnFirstFailure	Property used to gets or sets a value indicating whether to stop user input after the first invalid character is entered.

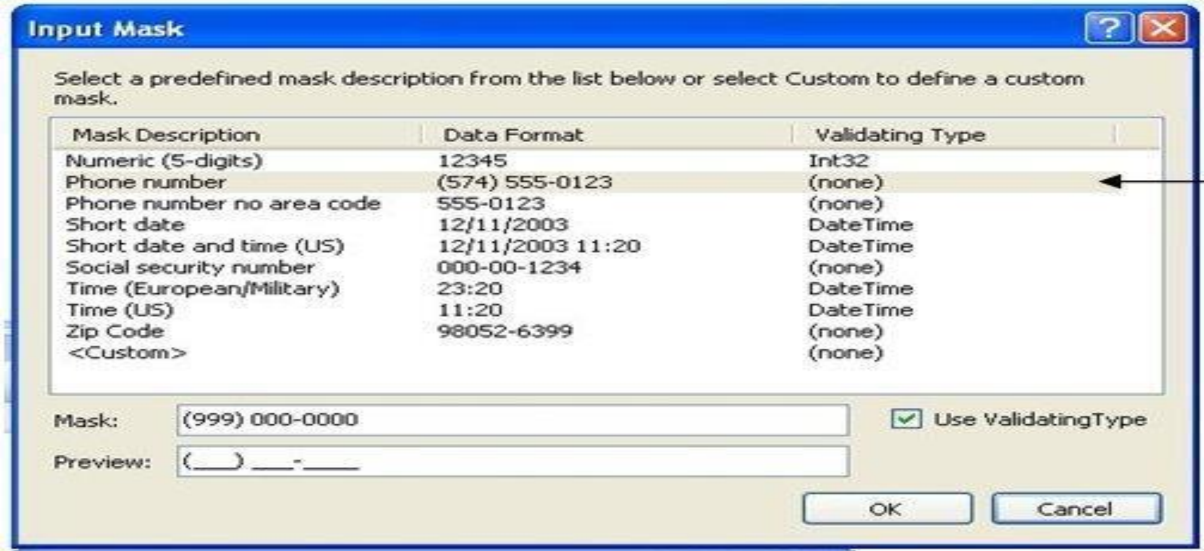


Properties

mskPhone System.Windows.Forms.MaskedTextBox

HideSelection	True
ImeMode	NoControl
InsertKeyMode	Default
⊞ Location	**140, 67**
Locked	False
⊞ Margin	3, 3, 3, 3
Mask	**(999) 000-0000**
⊞ MaximumSize	0, 0

Set the Mask property to phone number and this mask will appear like this in the property box.



Input Mask

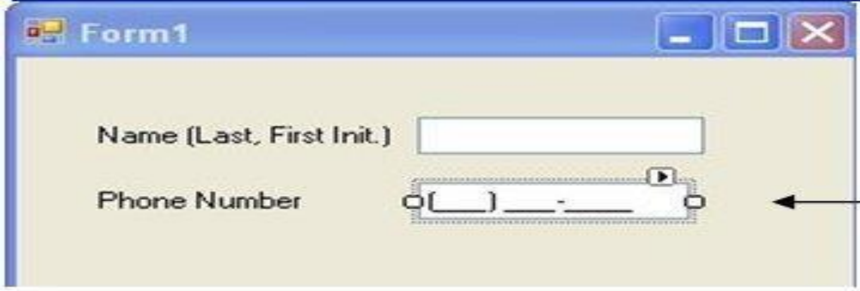
Select a predefined mask description from the list below or select Custom to define a custom mask.

Mask Description	Data Format	Validating Type
Numeric (5-digits)	12345	Int32
Phone number	(574) 555-0123	(none)
Phone number no area code	555-0123	(none)
Short date	12/11/2003	DateTime
Short date and time (US)	12/11/2003 11:20	DateTime
Social security number	000-00-1234	(none)
Time (European/Military)	23:20	DateTime
Time (US)	11:20	DateTime
Zip Code	98052-6399	(none)
<Custom>		(none)

Mask: (999) 000-0000 ☑ Use ValidatingType

Preview: (___) ___-____

Set the Mask property in the Input Mask dialog box, which appears when you click the ellipsis button on the right of the Mask property in the masked text box's Property box.



Form1

Name (Last, First Init.)

Phone Number (___) ___-____

The masked text box should look like this after you set its Mask property the phone number by following the above steps.

NotifyIcon Control

- Icons in the status area are short cuts to processes that are running in the background of a computer, such as a virus protection program or a volume control. These processes do not come with their own user interfaces. The NotifyIcon class provides a way to program in this functionality.
- **Icon** property defines the icon that appears in the status area.
- **ContextMenu** Pop-up menus for an icon are addressed with the **ContextMenu** property.
- **Text** property assigns ToolTip text. In order for the icon to show tool tip bring the mouse to the icon.

LinkLabel

- LinkLabel is similar to a Label but they display a hyperlink.
- Even multiple hyperlinks can be specified in the text of the control and each hyperlink can perform a different task within the application.
- They are based on the Label class which is based on the Control class.
- Notable properties of the LinkLabel control are the ActiveLinkColor, LinkColor and LinkVisited which are used to set the link color.

Timer

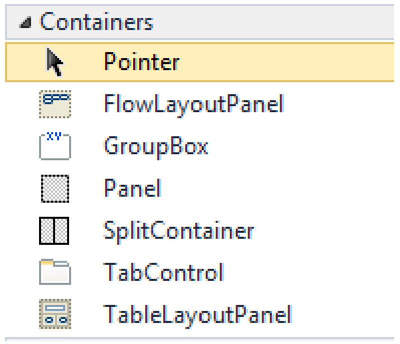
- A Timer is used to raise an event at user-defined intervals. It has the following important events.
  - **Interval** The number of milliseconds between each timer tick. The value is not less than one. It is in milliseconds.
  - **Enabled** It allows starting and stopping the timer by setting the value true or false.

Container control

Container controls, as their name implies, are controls that can **host other controls inside them**.

The Form is the perfect example here, as you put all your controls on the form. The host is known as the **parent**, and the controls inside the host are known as the **children**.

Container controls can obviously host other Container controls as well; then you'relooking at a **grandparent**, parent and child relationship



Panel, GroupBox

- Panels are those controls which contain other controls, for example, a set of radio buttons, checkboxes, etc.
- Panels are similar to Groupboxes but the difference, Panels cannot display captions where as GroupBoxes can and Panels can have scrollbars where as GroupBoxes can't.
- If the Panel's Enabled property is set to False then the controls which the Panel contains are also disabled. Panels are based on the ScrollableControl class.

- Notable property of the Panel control in the appearance section is the BorderStyle property.
- Notable property in the layout section is the AutoScroll property.