Windows Form Application

Windows Form:

- Windows Forms is a Graphical User Interface(GUI) class library which is bundled in .*Net Framework*.
- Its main purpose is to provide an easier interface to develop the applications for desktop, tablet, PCs. I
- t is also termed as the WinForms. The applications which are developed by using Windows Forms or WinForms are known as the Windows Forms Applications that runs on the desktop computer.
- WinForms can be used only to develop the Windows Forms Applications not web applications.
- WinForms applications can contain the different type of controls like labels, list boxes, tooltip etc.

Points To Remember:

- Namespace:

Windows Form Application	System.Windows.Forms
Form (Single Form)	System.Windows.Forms.Form

- Type of files with extension:

Code Behind File	Form1.cs
Design File	Form1.cs[Design]
Designer File	Form1.Designer.cs
Resource File	Form1.resx

- Everything in form is known as OBJECT.
- GUI Applications are based on EVENTS.
- Every control has a property and its associated events.
- Default event for Form : LOAD
- Default event for Button: CLICK

❖ Common Properties :(Related to Form and controls)

Property	Description	Example
Text	The text associated with the control.	Form1,chechbox1
Name	Indicates the name used in code to identify the object	Form1
Locked	Determines if we can move or resize the control	Default value: False
BackColor	Background color of the component	Custom/web/System Color
Background Image	BackgroundImage used for the control	Import from local resource or resource file
Font	Font used to display text in the control	Microsoft Sans Serif, 7.8pt
ForeColor	Foreground color of this component , which is used to display text.	Custom/web/System Color
ContextMenuStrip	The shortcut menu display when the user right clicks the control	if not set, then(none)
Enabled	Indicates whether the control is enabled.	Default: True
Auto Size	Specifies whether a control will automatically size itself to fit its content	Default : False for form and True for other control such as checkbox.
Auto Size Mode	Specifies the mode by which the user interface element automatically resize itself.	GrowOnly/GrowandShrink
Location	The coordinates of the upper left corner of the control relative to the upper left corner of its container.	0,0
MaximumSize	Form: Maximum size of the form can be resized to. Controls: Specifies the maximum size of the control.	0,0
MinimumSize	Form: Minimum size of the form can be resized to. Controls: Specifies the minimum size of the control.	0,0
Padding	Specify the interior spacing of the control.	0,0,0,0
Size	Size of the control in pixels. (height,width)	300,300
StartPosition	Determines the position of a form when it first appears.	Default:WindowsDefaultLocation

		Other options: Manual, CenterScreen, WindowsDefaultBounds, CenterParent
WindowState	Determines the initial visual state of the form	Normal/Maximized/Minimized.
Icon	Indicates the icon for a form.This icon is displayed in the form’s system menu box and when the form is minimized.	Icon(image)
IsMdiContainer	Determines whether the form is an MDI container.	False/True
ShowIcon	Indicates whether an icon is displayed in the title bar of the form.	True/False
Dock	Defines which borders of the control are bound to the container.	Left/right/top/ bottom/none
ToolTip on toolTip1	Determines the ToolTip shown when the mouse hovers over the control.	Enter name/hello/etc.

- **Common Events :(Related to Form and controls)**

Event	Description
Click	Occurs when the component is clicked.
Load	Occurs whenever the user loads the form
TextChanged	Event raised when the value of the Text Property is changed on Control.
Enter	Occurs when the control becomes the active control of the form.
Leave	Occurs when the control is no longer th active form of the control.
KeyDown	Occurs when a key is first pressed.
KeyPress	Occurs when the control has focus and the user presses and releases a key.
KeyUp	Occurs when a key is released.
DoWork	Event handler to be run on a different threadwhen the operation begins.
HelpRequest	Occurs when the user clicks the Help Button.

❖ **Controls:**

Name	Description	Property	Namespace	Event	Syntax (Example)

Label	Provides run-time information or descriptive text for a control.	Autosize, font, text, textalign, borderstyle, etc.	System.Windows.Forms .Label	Default: click
				Others: Leave, Lostfocous, TextChanged
					private void label1_Click(object sender, EventArgs e)
					{
					this.BackColor = Color.Red;
					}

TextBox	Enables the user to enter text, and provides multiline editing and password character masking.	PasswordChar, Multiline, Autosize, MaxLength, WordWrap, ReadOnly, Visible,etc.	System.Windows.Forms .TextBox	Default: TextChanged
				Others: TextAlignChanged, click
					private void textBox1_TextChanged(object sender, EventArgs e)
					{
					textBox1.Text = "Hello";
					}

Masked TextBox	Uses a mask to distinguish between proper and improper user input.	Mask, PromptChar, TextMaskFormat, Multiline, ReadOnly, BeepOnError, etc.	System.Windows.Forms .MaskedTextBox	Default: MaskInputRejected Others: MaskChanged, TextChanged	private void maskedTextBox1_MaskInputRejected(object sender, MaskInputRejectedEventArgs e) { maskedTextBox1.Mask = "00/00/0000";}

Button	Raises an event when the user clicks it.	Image, ImageAlign, Enable, Size, Font, FlatStyle, DialogResult, etc.	System.Windows.Forms .Button	Default: click
				Others: Validated, TextChanged, FontChange, GotFocous
					private void button1_Click(object sender, EventArgs e)
					{
					button1.Text = "Submit";
					}

RadioButton	Enables the user to select a single option from a group of choices	Checked, Image, CheckAlign, BackgroundImage,	System.Windows.Forms .RadioButton	Default: CheckedChanged Others:	private void radioButton1_CheckedChanged(object sender, EventArgs e) {

	when paired with other RadioButtons.	BackgroundImageLayout		Click,
				TextChanged
					MessageBox.Show("Hello");
					}

CheckBox	Enables the user to select or clear the associated option.	Checked, CheckSate (Unchecked/ Checked/ Intermediate), Image, BackgroundImage, CausesValidation (Focous)	System.Windows.Forms.CheckBox	Default:
				CheckedChanged
				
				Others:
				CheckStateChanged,
				Click,
				TextChanged,
				EnableChanged.
					private void checkBox1_CheckedChanged(object sender, EventArgs e)
					{
					label1.Text = checkBox1.Text;
					}

ListBox	Displays a list from which the user can select items.	Items(Collection), SelectionMode (One/MultiSimple/ MultiExtended), Sorted, Drawmode	System.Windows.Forms.ListBox	Default:
				SelectedIndexChanged
				
				Others:
				StyleChanged,
				Click
					private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
					{
					label1.Text = listBox1.SelectedItem.ToString();
					}

ComboBox	Display an editable text box with a drop-down list of permitted values.	Items(Collection),
		DropDownHeight,
		DropDownWidth,
		TabIndex
			System.Windows.Forms.ComboBox	Default:
				SelectedIndexChanged
				Others:
				SelectionChangeCommitted,
				Click, Leaved
					private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
					{
					label1.Text = comboBox1.SelectedItem.ToString();
					}

GroupBox	Displays a frame around a group of control with an optional caption.	UseCompatibleTextRendering	System.Windows.Forms.GroupBox	Default:
				Enter
				Others:
				Leave,Validated
					private void groupBox1_Enter(object sender, EventArgs e)
					{
					MessageBox.Show("Hii");
					}

PictureBox	Displays an image.	ErrorImage,
		ImageLocation,
		InitialImage,
		WaitOnLoad
			System.Windows.Forms
			.PictureBox
				Default:
				Click
				Others:
				Validated,
				Paint,DoubleClick
					private void pictureBox1_Click(object sender, EventArgs e)
					{
					MessageBox.Show("hii");
					}

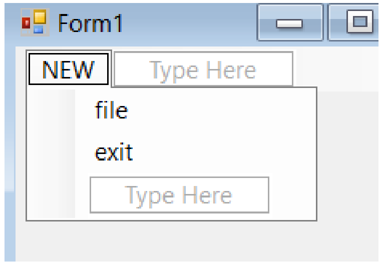
LinkLabel	Displays a label control that supports hyperlink functionality, formatting, and tracking.	ActiveLinkColor,
		DisabledLinkColor,
		LinkColor,
		LinkVisited,
		Image
			System.Windows.Forms
			.LinkLabel
				Default:
				LinkClicked
				Others:
				Click,
				TextChanged
					private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
					{
					MessageBox.Show("hii");
					}

ToolTip
(ToolTip on toolTip1)
	Displays information when the user moves the pointer over an associated control.	Active,
		InitialDelay,
		AutomaticDelay,
		IsBallon,
		ReshowDelay,
		ToolTipIcon,
		ToolTipTitle,
		AutoPopDelay
			System.Windows.Forms
			.ToolTip
				Default:
				Popup
				Others:
				Draw
					private void toolTip1_Popup(object sender, PopupEventArgs e)
					{
					MessageBox.Show("hii");
					}

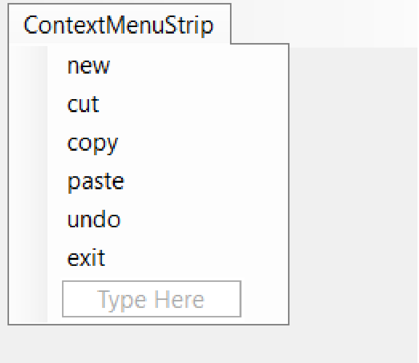
DateTime Picker	Enables the user to select a date and time, and to display that date and time in a specified format.	Value,
		MaxDate,
		MinDate,
		AllowDrop,
		CustomFormat,
		Checked
			System.Windows.Forms
			.DateTimePicker
				Default:
				ValueChanged
				Others:
				CloseUp,
				DropDown,
				MouseCapturedChanged
					private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
					{
					MessageBox.Show(dateTimePicker1.Value.ToString());
					}

MenuStrip	Displays application commands and options grouped by	Items, MdiWindowListItem, AllowMerge,	System.Windows.Forms .MenuStrip	Default: ItemClicked Others:	private void menuStrip1_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

	functionality.	AllowDrop,
		AllowItemReorder,
		ShowItemToolTips
				MouseClick,
				Click
					{
					MessageBox.Show("Hii");
					}

ToolStripMenuItem	Shows Menu Item (create a menu bar with shortcut key)	DropDown,
		DropDownItems,
		ShortcutKeys,
		ShowShortcutKeys,
		CheckOnClick,
		AutoToolTip,
		ToolTipText
			System.Windows.Forms.ToolStripMenuItem	Default:
				Click
				
				Others:
				DropDownItemClicked,
				DropDownClosed,
				DropDownOpened,
				DropDownOpening,
				DropDownOpened,
				CheckChanged,
				CheckStateChanged,
				DisplayStyleChanged
					private void nEWToolStripMenuItem_Click(object sender, EventArgs e)
					{
					
					}
					

ContextMenuStrip	Displays a shortcut menu when the user right-clicks the associated control.	Items,
		ShowItemToolTips,
		DropShadowEnabled
		
		Tasks:
		RenderMode,
		ShowImageMargin,
		ShowCheckMargin
			System.Windows.Forms.ContextMenuStrip	Default:
				Opening
				
				Others:
				Click,
				Closed, Opened,
				ItemClicked,
				MouseClick,
				RendererChanged
					private void contextMenuStrip1_Opening(object sender, CancelEventArgs e)
					{
					MessageBox.Show("Hello");
					}

					 ContextMenuStrip new cut copy paste undo exit Type Here

ProgressBar	Displays a bar that fills to indicate to the user the progress of an operation.	Value, Maximum, Minimum, MarqueeAnimationSpeed, RighttoLeft, Step, Style, UseWaitCursor	System.Windows.Forms.ProgressBar	Default: Click
				
				Others: MouseClick
					private void progressBar1_Click(object sender, EventArgs e)
					{
					MessageBox.Show("Hii");
					}

DataGridView	Displays rows and columns of data in a grid you can customize.	MultiSelect, ReadOnly, SelectionMode, VirtualMode, GridColor, ShowCellToolTips	System.Windows.Forms.DataGridView	Default: CellContentClick
				
				Others: CellClick, DataError, CellMouseClick
					private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
					{
					
					}

RichTextBox	Provides advanced text entry and editing features such as character and paragraph formatting.	Text, Lines, RightToLeft, ScrollBars, WordWrap, ZoomFactor, MultiLine, MaxLength,	System.Windows.Forms.RichTextBox	Default: TextChanged
				
				Others: Click
					private void richTextBox1_TextChanged(object sender, EventArgs e)
					{
					MessageBox.Show("Hii");
					}

		HideSelection			

CheckedList Box	Displays a list of items with a check box on the left side of each item.	Items,
		SelectionMode,
		Sorted,
		MultiColumn
			System.Windows.Forms .CheckedListBox	Default:
				SelectedIndexChanged
				
				Others:
				Click,
				ItemCheck,
				MouseClick
					private void checkedListBox1_SelectedIndexChanged(object sender, EventArgs e)
					{
					MessageBox.Show("Hii");
					}

HScrollBar	Enables its parent component to scroll content horizontally. Unnecessary if the parent has an autoscroll attribute, and that attribute is enabled.	Value,
		Maximum,
		Minimum,
		LargeChange,
		SmallChange,
			System.Windows.Forms .HScrollBar	Default:
				Scroll
				
				Others:
				ValueChanged,
				MouseCaptureChanged
					private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
					{
					MessageBox.Show("Hii");
					}

VScrollBar	Enables its parent component to scroll content vertically. Unnecessary if the parent has an autoscroll attribute, and that attribute is enabled.	Value,
		Maximum,
		Minimum,
		LargeChange,
		SmallChange,
			System.Windows.Forms .VScrollBar	Default:
				Scroll
				
				Others:
				ValueChanged,
				MouseCaptureChanged
					private void vScrollBar1_Scroll(object sender, ScrollEventArgs e)
					{
					MessageBox.Show("Hii");
					}

Timer	Component that raises an event at user defined intervals.	Interval, Enabled, GenerateMember, Modifiers	System.Windows.Forms .Timer	Default: Tick	private void timer1_Tick(object sender, EventArgs e) {

					}

BackgroundWorker	Executes an operation on a separate thread.	WorkerReportsProgress,
		WorkerSupportsCancellation,
		GenerateMember,
		Modifiers
			System.Windows.Forms.BackgroundWorker	Default:
				DoWork
				
				Others:
				ProgressChanged,
				RunWorkerCompleted
					private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
					{
					
					}

- ❖ Dialog:
- A Dialog Box is a type of windows, which is used to enable common communication or dialog between a computer and its user.
- A Dialog Box is most often used to provide the user with the means for specifying how to implement a command or to respond to a question.
- Dialog Box are special forms that are non-resizable.
- Form.ShowDialog Method:
  - ✓ Namespace: Systems.Windows.Forms
  - ✓ Assembly: System.Windows.Forms.dll
  - ✓ ShowDialog( ):
    - Shows the form as a modal dialog box.
    - Returns: Dialog result (One of the DialogResult values).
    - Exceptions: InvalidOperationException

Name	Description	Property	Namespace	Event	Syntax (Example)

SaveFileDialog	Displays a dialog box that prompts the user to select a location for saving a file.	FileName,
		InitialDirectory,
		Filter,
		RestoreDirectory
			System.Windows.Forms.SaveFileDialog	Default:
				FileOk
				
				Others:
				HelpRequest
					DialogResult res = saveFileDialog1.ShowDialog();
					if(res == DialogResult.OK)
					{System.IO.File.WriteAllText(saveFileDialog1.FileName, textBox1.Text.ToString()); }

OpenFileDialog	Displays a dialog box that prompts the user to open a	FileName,
		InitialDirectory,
		Filter,
			System.Windows.Forms.OpenFileDialog	Default:
				FileOk
					DialogResult res = openFileDialog1.ShowDialog();
					if(res == DialogResult.OK)

	file.	MultiSelect,
		ReadOnlyChecked
		ShowReadOnly
		RestoreDirectory
				Others:
				HelpRequest
					{
					System.IO.File.WriteAllText(openFileDialog1.FileName, textBox1.Text.ToString());
					}

ColorDialog	Displays available colors along withcontrols that enable the user to define custom colors.	Color,
		FullOpen,
		AnyColor,
		ShowHelp,
		SolidColorOnly,
		AllowFullOpen
			System.Windows.Forms.ColorDialog	HelpRequest	DialogResult res = colorDialog1.ShowDialog();
					if(res == DialogResult.OK)
					{ this.BackColor = colorDialog1.Color;
					}

FontDialog	Displays a dialog box that prompts the user to choose a font from those those installed on the local computer.	Font, MaxSize,
		MinSize,
		ScriptsOnly,
		FontMustExist
			System.Windows.Forms.FontDialog	Default:
				Apply
				
				Others:
				HelpRequest
					DialogResult res = fontDialog1.ShowDialog();
					if(res == DialogResult.OK)
					{
					this.label1.Font=fontDialog1.Font;
					}

- Other Dialog Box available are: FolderBrowserDialog, PageSetupDialog, PrintDialog, PrintPreviewDialog.

❖ **MessageBox:**

- Namespace: System.Windows.Forms
- Assembly: System.Windows.Forms.dll
- Displays a message window, also known as a dialog box, which presents a message to the user. It is a modal window, blocking other actions in the application until the user closes it. A MessageBox can contain text, buttons, and symbols that inform and instruct the user.
- Example:

```
      result = MessageBox.Show(message, caption, buttons);
     if (result == System.Windows.Forms.DialogResult.Yes)
     {
            this.Close();
        }
```

- You cannot create a new instance of the MessageBox class. To display a message box, call the static method MessageBox.Show. The title, message, buttons, and icons displayed in the message box are determined by parameters that you pass to this method.

- Methods:

Show(String)	Displays a message box with specified text.
Show(String, String)	Displays a message box with specified text and caption.
Show(String, String, MessageBoxButtons)	Displays a message box with specified text, caption, and buttons.
Show(String, String, MessageBoxButtons, MessageBoxIcon)	Displays a message box with specified text, caption, buttons, and icon.
Show(String, String, MessageBoxButtons, MessageBoxIcon, MessageBoxDefaultButton, MessageBoxOptions, String, String)	Displays a message box with the specified text, caption, buttons, icon, default button, options, and Help button, using the specified Help file and Help keyword.
ToString()	Returns a string that represents the current object. (Inherited from Object)

❖ Form Redirect (Navigation between Forms):

- We can redirect from one form to another by creating instance.

```
private void button1_Click(object sender, EventArgs e)
{
    Form2 ob1 = new Form2();
    ob1.Show();
}
```

❖ MDI Form:

- Multiple-document interface (MDI) applications enable you to display multiple documents at the same time, with each document displayed in its own window. MDI applications often have a Window menu item with submenus for switching between windows or documents.

- MDI Form are also known as Menu Driven Interface Form.

- Any windows can become an MDI parent, if you set the IsMdiContainer property to True.
  Syntax: IsMdiContainer = true;
- Example:

```
private void Form1_Load(object sender, EventArgs e)
    {
        IsMdiContainer = true;
    }

    private void menu1ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Form2 frm2 = new Form2();
        frm2.Show();
        frm2.MdiParent = this;
    }
```

❖ Can we have Multiple form within application?

- ✓ Yes, We can have multiple form within same windows form application.
- ✓ Solution Explorer → Project Name → Right Click → Add → New Item →Add the form (form2.cs)

❖ Set property dynamically:

- ✓ We can set the property dynamically using "this" keyword.
- ✓ Example:

```
this.BackColor = colorDialog1.Color;
this.Enabled =true;
```

❖ Difference between

• ListBox and ComboBox

- ✓ The difference between a List box and a Combo box is that a List box is simply a list of items, while a Combo box is a combination of a List box and an Edit box.

- ✓ A List box allows the user to choose multiple selections from the list at one time, while a Combo box allows only a single selection, but a Combo box allows the user to edit the selected value while a List box only allows the user to choose from the given list.

• TextBox and RichTextBox

- ✓ A RichTextBox mainly used if you want more control over styling the text color, type, font, alignment ect. So anything you can do in Microsoft Word, you can do with a RichTextBox. It can be used to save or display .rtf files with ease.

- ✓ A Textbox is basically used to display or get one line input. You can have a multi-line TextBox which is used mainly to display or get more than one one-liner and keeps you from having to manage multiple TextBox's. Also keeps your UI a little more tidy.

- ✓ So basically the main difference is in styling. If you just want something plain and simple, use TextBox. If you want something fancy, eg styles, colors use a RichTextBox.

- **TextBox and MaskedTextBox**
  - ✓ Masked box is used for specific formats like dateformat or phone number etc while textbox is used for normal input with option for multiline.

- **MenuStrip and ContextMenuStrip**
  - ✓ The difference between a MenuStrip control and a ContextMenuStrip control is that a MenuStrip control is associated with the Windows Form; whereas, a ContextMenuStrip control is associated with a control, which is added to the Windows Form.

- **ListBox and CheckBox**
  - ✓ The difference between a ListBox control and CheckBox Control is that we can set multiple (different) properties for checkboxes while for ListBox we can set only property once.

❖ **Full Forms:**

- ○ GUI: Graphical User Interface
- ○ MDI: Menu Driven Interface OR Multiple Document Interface.
- ○ IME: Input Method Editor

❖ **Reference:**

- www.google.com
- www.geeksforgeeks.com
- www.javatpoint.com
- https://docs.microsoft.com/en-us/dotnet/csharp/