

Unit 4 & 5

Advanced GUI controls

Unit Covered

- Dialog Box:
 - Color, font, file open, file save and common dialog boxes
- RichTextBox:
 - Text manipulation and formatting
- Treeview control:
 - Adding nodes at design time and runtime, scanning tree view control
- ListView control:
 - The column collection, ListView Items and sub items, Items collection, Sub items collection, sorting in ListView, processing selected Items

Rich Text Boxes

- RichTextBoxes are similar to Textboxes but they provide some advanced features over the standard Textbox.
- RichTextBox allows formatting the text, say adding colors, displaying particular font types and so on.

Properties of Rich Text Box

Property	Means
AutoSize	Sets/gets a value specifying if the size of the rich text box automatically adjusts when the font changes.
BorderStyle	Sets/gets the border type of the rich text box.
BulletIndent	Sets/gets the indentation used in the rich text box when the bullet style is applied to the text.
Lines	Sets/gets the lines of text in a RichTextBox control.
Multiline	Sets/gets a value specifying if this is a multiline RichTextBox control.
ReadOnly	Sets/gets a value specifying if text in the rich text box is read-only.
ScrollBars	Sets/gets the kind of scroll bars to display in the RichTextBox control.

Events of Rich Text Box

Events	Means
TextChanged	Occurs when text is changed
Click	Occurs when the text box is clicked.
ReadOnlyChanged	Occurs when the value of the ReadOnly property is changed.

Methods of RichTextBox

Method	Means
AppendText	Appends text to the current text of the rich text box.
CanPaste	Determines if you can paste information from the clipboard.
Clear	Clears all text from the RichTextBox Control.
Copy	Copy the current selection in the rich text box to the clipboard.
Cut	Moves the current selection in the rich text box to the clipboard.
LoadFile	Loads the contents of a file into the RichTextBox control.
SaveFile	Saves the contents of the rich text box to a file.

Dialog Box

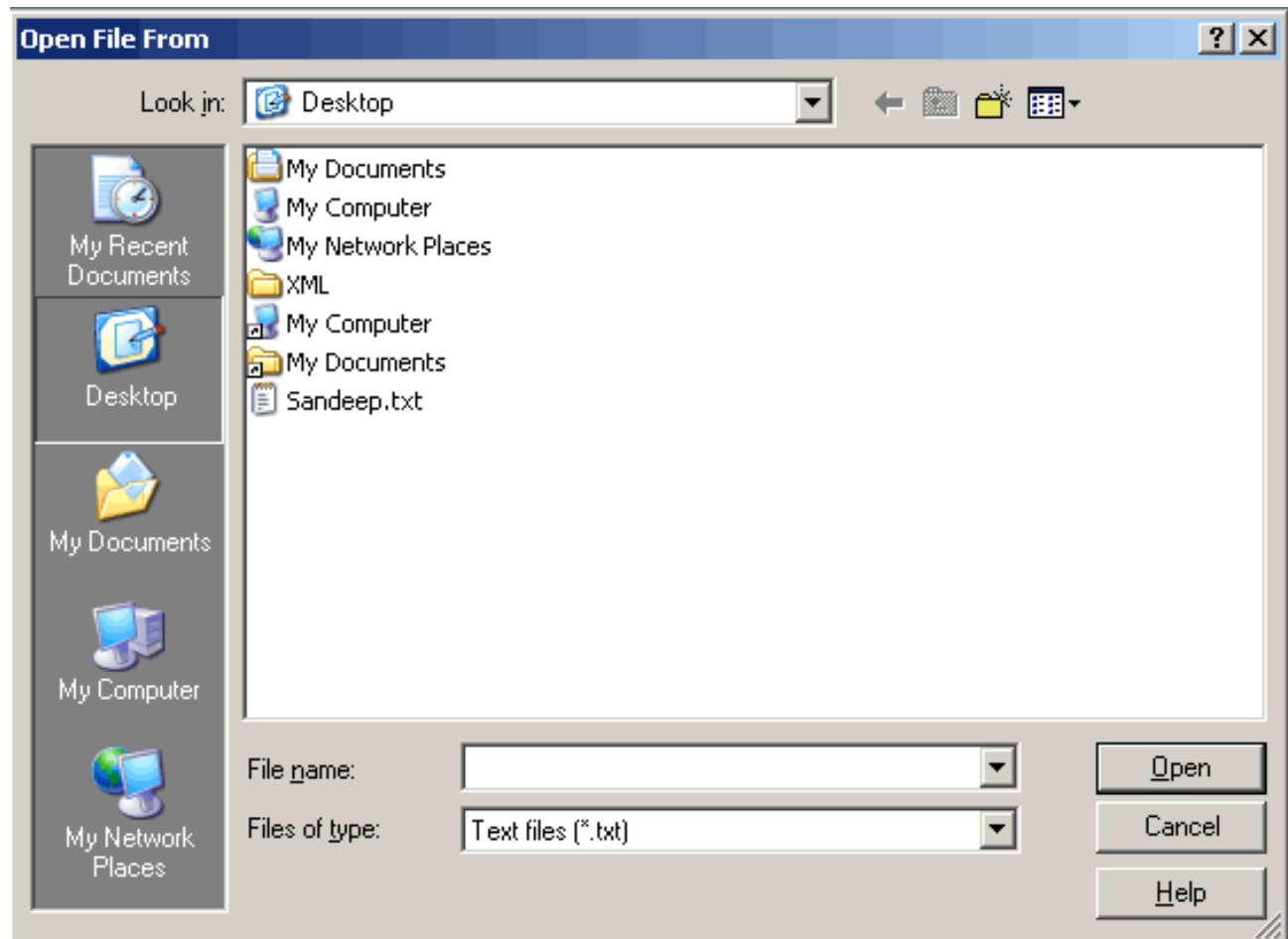
- Visual Basic .NET comes with built-in dialog boxes which allow us to create our own File Open, File Save, Font and Color dialogs.
- **Dialog Box :**
 - OpenFileDialog
 - SaveFileDialog
 - ColorDialog
 - FontDialog
 - PrintDialog,
 - PrintPreviewDialog
 - PageSetupDialog.

Dialog Box

- To make a dialog box visible at run time we use the dialog box's **ShowDialog** method.
- The return values of all the above said dialog boxes which will determine which selection a user makes are:
 - Abort
 - Cancel
 - Ignore
 - No
 - None
 - OK
 - Return
 - Retry
 - Yes.

OpenFileDialog Box

- Open File Dialog's are supported by the OpenFileDialog class .
- They allow us to select a file to be opened.



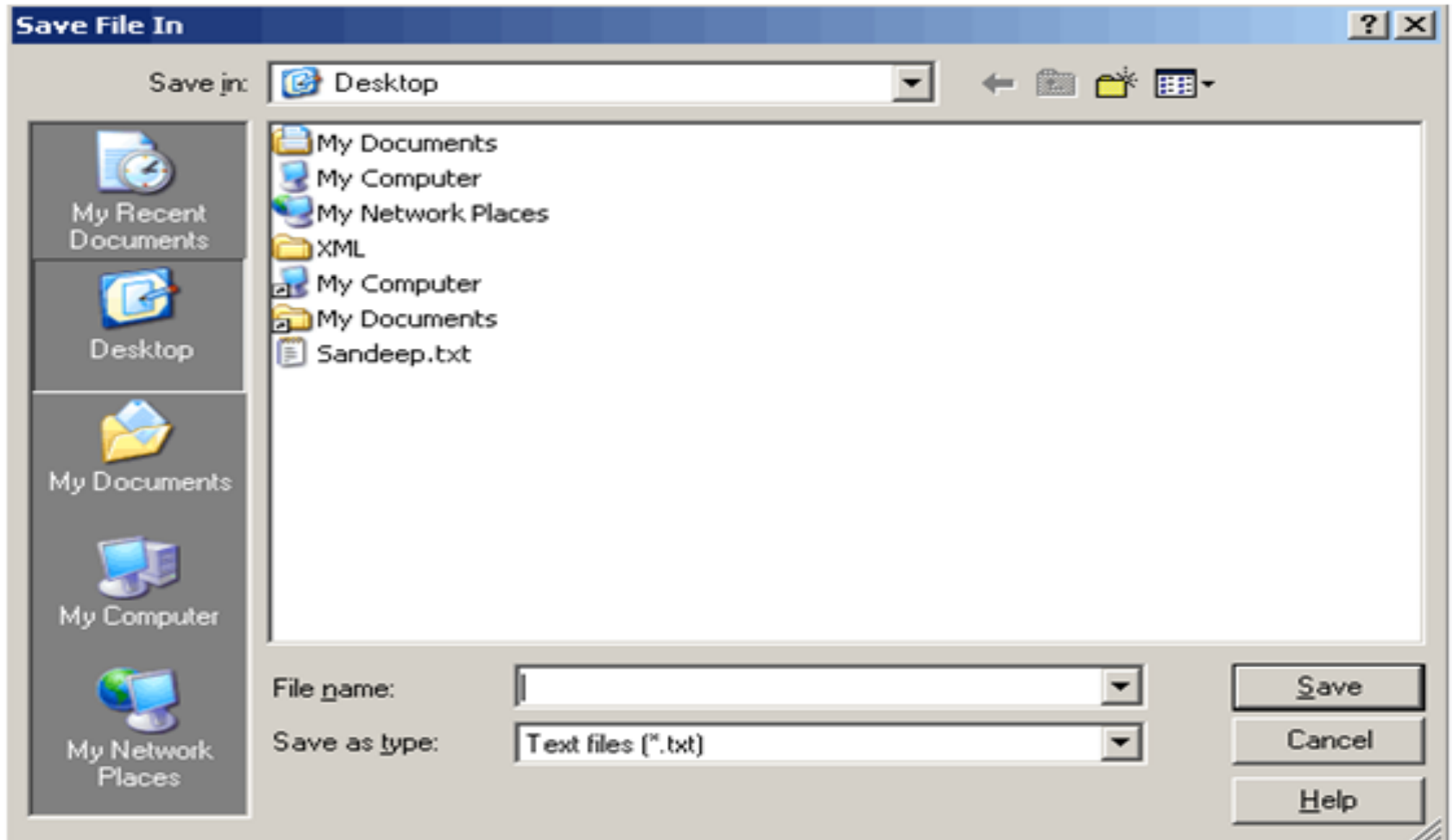
OpenFileDialog Box

Properties	Mean
AddExtension	Gets/Sets if the dialog box adds extension to file names if the user doesn't supply the extension.
CheckFileEixsts	Checks whether the specified file exists before returning from the dialog.
CheckPathExists	Checks whether the specified path exists before returning from the dialog.
DefaultExt	Allows you to set the default file extension
FileName	Gets/Sets file name selected in the file dialog box
FileNames	Gets the file names of all selected files.
Filter	Gets/Sets the current file name filter string, which sets the choices that appear in the "Files of Type" box.

SaveFileDialog

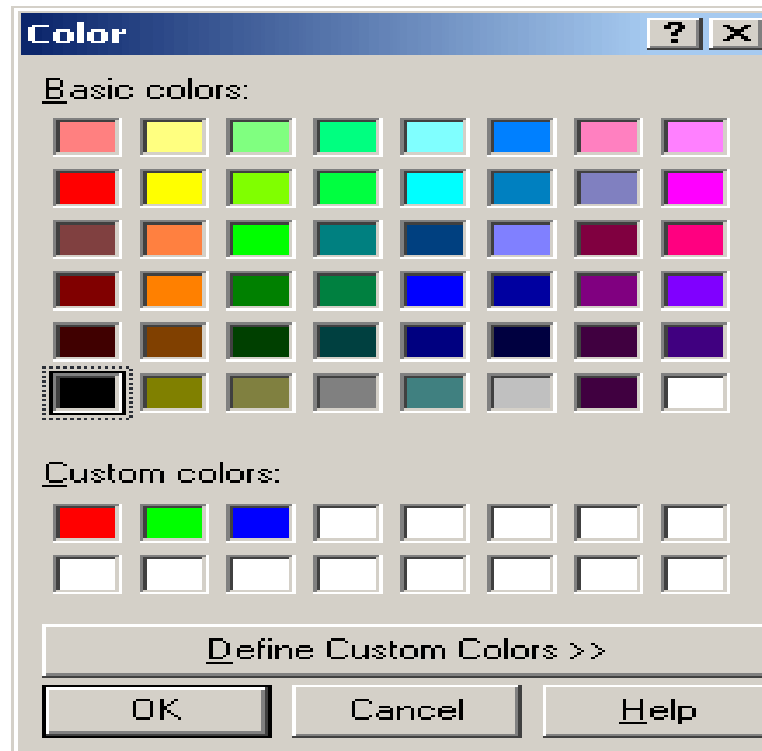
- Save File Dialog's are supported by the SaveFileDialog class and they allow us to save the file in a specified location.
- Properties of the Save File Dialog are the same as that of the Open File Dialog.
- Notable property of Save File dialog is the **OverwritePromopt** property which displays a warning if we choose to save to a name that already exists.

SaveFileDialog



ColorDialog

- Color Dialog's are supported by the ColorDialog Class and they allow us to select a color. The image below displays a color dialog.

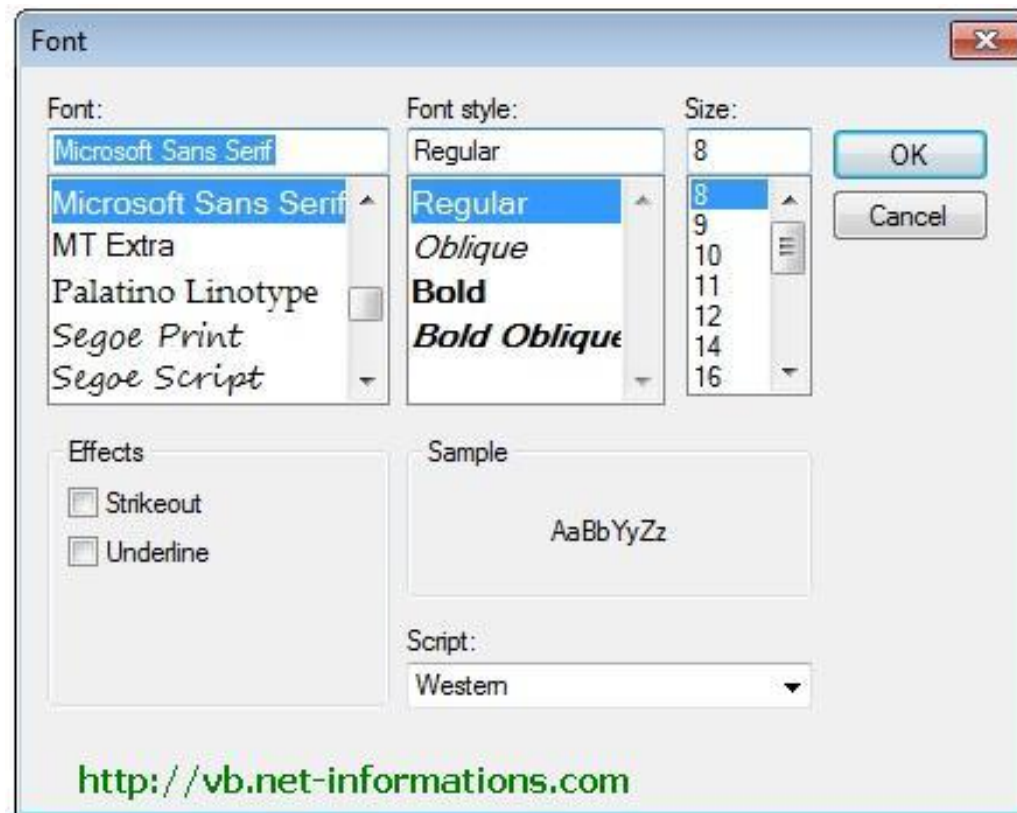


ColorDialog box properties

Property	Meaning
AllowFullOpen:	Gets/Sets whether the user can use the dialog box to define custom colors.
AnyColor:	Gets/Sets whether the dialog box displays all the available colors in the set of basic colors.
Color:	Gets/Sets the color selected by the user.
CustomColors:	Gets/Sets the set of custom colors shown in the dialog box.
ShowHelp:	Gets/Sets whether the dialog box displays a help button.
SolidColorOnly:	Gets/Sets whether the dialog box will restrict users to selecting solid colors only.

FontDialog Box

- The Font dialog box lets the user choose attributes for a logical font,
 - font family, font style, point size, effects , and a script .

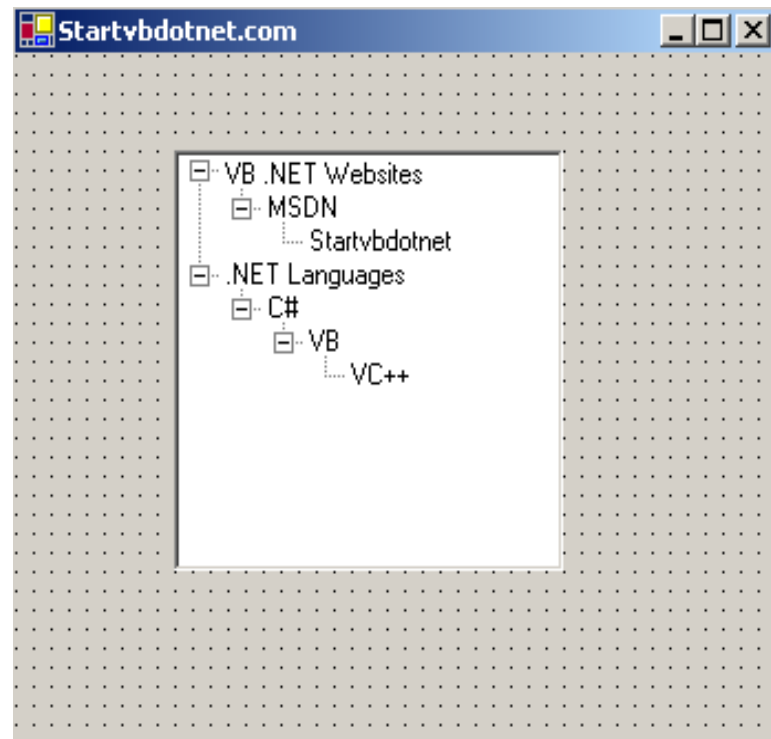


FontDialog Box

Properties	Meaning
AllowVerticalFonts:	Gets/Sets whether the dialog box displays both vertical and horizontal fonts or only horizontal fonts.
Color:	Gets/Sets selected font color.
Font:	Gets/Sets the selected font.
MaxSize:	Gets/Sets the maximum point size the user can select.
MinSize:	Gets/Sets the mainimum point size the user can select.
ShowColors:	Gets/Sets whether the dialog box displays the color choice.
ShowEffects	Gets/Sets whether the dialog box contains controls that allow the user to specify to specify strikethrough, underline and text color options.
ShowHelp:	Gets/Sets whether the dialog box displays a help button.

TreeView

- The tree view control is used to display a hierarchy of nodes (both parent, child).
- You can expand and collapse these nodes by clicking them.



Properties of TreeView

Property	Means
BorderStyle	Gets/sets the tree view's border style.
CheckBoxes	Gets/sets whether checkboxes should be displayed next to tree nodes.
Nodes	Gets the collection of tree nodes.
SelectedNode	Gets/sets the node that is selected.
TopNode	Gets the first visible tree node.
Sorted	Gets/sets if the tree nodes should be sorted.
ShowPlusMinus	Gets/sets whether plus-sign (+) and minus-sign (-) buttons are shown next to tree nodes with child tree nodes.
LastName:	Gets the last child node
NextNode:	Gets the next sibling node

Methods of TreeView

Method	Means
BeginUpdate	Disables redrawing of the tree view.
CollapseAll	Collapses all nodes.
EndUpdate	Enables redrawing of the tree view.
ExpandAll	Expands all the nodes.
GetNodeAt	Gets the node that is at the given location.
GetNodeCount	Gets the number of nodes.

Events of TreeView

Event	Means
AfterCheck	Occurs when a node checkbox is checked.
AfterCollapse	Occurs when a tree node is collapsed.
AfterExpand	Occurs when a tree node is expanded.
AfterLabelEdit	Occurs when a tree node label text is edited.
AfterSelect	Occurs when a tree node is selected.
BeforeCheck	Occurs before a node checkbox is checked.
BeforeCollapse	Occurs before a node is collapsed.
BeforeExpand	Occurs before a node is expanded.
BeforeLabelEdit	Occurs before a node label text is edited.
BeforeSelect	Occurs before a node is selected.
ItemDrag	Occurs when an item is dragged into the tree view.

Adding node at runtime

- **How to add a Root Node**

```
TreeView1.Nodes.Add("Node Name")
```

```
TreeView1.Nodes.Add(txtNode.Text)
```

```
TreeView1.Nodes.Add(InputBox("ENTER NODE NAME"))
```

- **How to add a Child node to a selected node**

```
TreeView1.SelectedNode.Nodes.Add("Node Name")
```

```
TreeView1. SelectedNode.Nodes.Add(txtNode.Text)
```

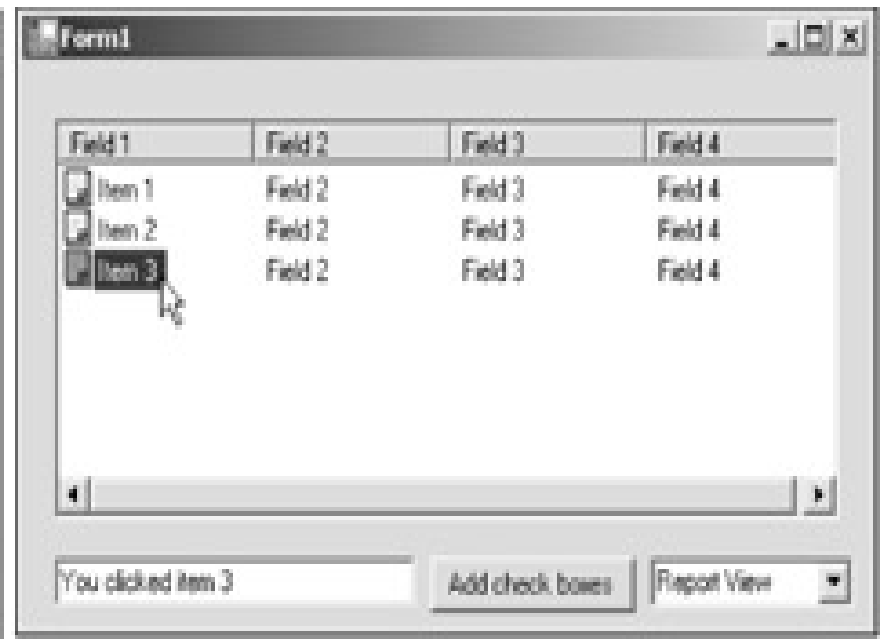
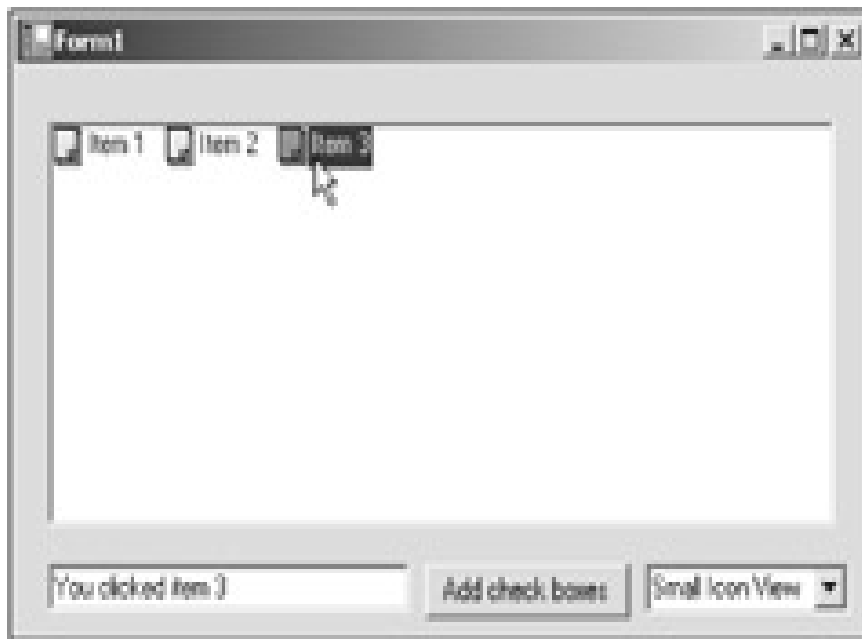
```
TreeView1.SelectedNode.Nodes.Add("InputBox("ENTER NODE  
NAME"))")
```

ListView control:

- The column collection,
- ListView Items and sub items
- Items collection
- Sub items collection
- Sorting in ListView
- Processing selected Items

ListView

- If tree views are all about displaying node hierarchies, like the folder hierarchy on a disk,
- list views are all about displaying lists of items.



- List views can display their items in four *view modes*
 - **View.LargeIcon**
 - The large icon mode displays large icons (large icons are 32×32 pixels) next to the item text.
 - **View.SmallIcon**
 - The small icon mode is the same except that it displays items using small icons (small icons are 16×16 pixels).
 - **View.List**
 - The list mode displays small icons, always in one column
 - **View.Details**
 - The report mode (also called the details mode) displays items in multiple columns

Property

- **ListItems** :which contains the items displayed by the control.
- **SelectedItems** : contains a collection of the items currently selected in the control.
- **MultiSelect** : The user can select multiple items if the **MultiSelect** property is set to **True**.
- **CheckBoxes** :list views can display checkboxes next to the items, if the **CheckBoxes** property is set to **True**.
- **Sorting**: Sort Item in listbox

Property

- **Activation** property sets what action the user must take to activate an item in the list:
 - **OneClick** requires a single click to activate the item.
 - **TwoClick** requires the user to double-click (a single click changes the color of the item text).
 - **Standard** requires the user to double-click to activate an item (but in this case, the item does not change appearance).

Methods of Listview

Method	Means
ArrangeIcons	Arranges the displayed items in Large Icon or Small Icon view.
BeginUpdate	Stops the list view from redrawing.
Clear	Removes all items from the list view.
EndUpdate	Allows redrawing of the list view.
EnsureVisible	Makes sure that an item is visible.
GetItemAt	Gets the item corresponding to the given X,Y coordinate.

Events

Events	Means
AfterLabelEdit	Occurs when a label has been edited.
BeforeLabelEdit	Occurs before a label is changed.
ColumnClick	Occurs when a column is clicked.
ItemActivate	Occurs when an item is activated.
ItemCheck	Occurs when an item is checked.
SelectedIndexChanged	Occurs when the selected index changes.

ListView Item Property

- The items in a list view are objects of the **ListViewItem** class, and the collection of those items is stored in the list view's **Items** property.

Property	Means
Bounds	Gets the bounding rectangle of an item, including its subitems.
Checked	True if the item is checked, False otherwise.
Index	Gets the index in the list view of the item.
ListView	Gets the list view that contains this item.
Selected	Gets/sets if the item is selected.
SubItems	Gets a collection of the subitems of this item.
Text	Gets the text for this item.

Panel Control & GroupBox

- **GroupBoxes and Panels**

- **Arrange controls on a GUI**
- **Used to group similar functionality that is related**
- **Primary difference between these two controls:**
 - **GroupBoxes can display a caption (i.e., text) and do not include scrollbars**
 - **Panels can include scrollbars and do not include a caption**

GroupBox properties	Description
Controls	The set of controls that the GroupBox contains.
Text	Specifies the caption text displayed at the top of the GroupBox.

Panel properties	Description
AutoScroll	Indicates whether scrollbars appear when the Panel is too small to display all of its controls. The default value is false.
BorderStyle	Sets the border of the Panel. The default value is None; other options are Fixed3D and FixedSingle.
Controls	The set of controls that the Panel contains.

ToolTip

Helpful text that appears when the mouse hovers over an item

ToolTip properties and events	Description
<i>Common Properties</i>	
AutoPopDelay	The amount of time (in milliseconds) that the tool tip appears while the mouse is over a control.
InitialDelay	The amount of time (in milliseconds) that a mouse must hover over a control before a tool tip appears.
ReshowDelay	The amount of time (in milliseconds) between which two different tool tips appear (when the mouse is moved from one control to another).
<i>Common Event</i>	
Draw	Raised when the tool tip is displayed. This event allows programmers to modify the appearance of the tool tip.

ImageList Control

An ImageList is a supporting control that is typically used by other controls, such as a ListView but is exposed as a component to developers. We can use this component in our applications when we are building our own controls such as a photo gallery or an image rotator control.

Properties	Description
ColorDepth	Property gets the color depth for this image list.
Handle	Property gets the handle for this image list.
Images	Property gets an ImageCollection object for this image list.
ImageSize	Property used to get or set the image size for the images in the list.
TransparentColor	Property used to get or set the transparent color for this list.

Methods:

Method	Description
Draw	Method used to draw the given image.

SplitContainer Control

- SplitContainer is an advance window's control or it is an extended version of old splitter control. You can place it at form horizontally or vertically by specifying Orientation property each represents left/top and right/bottom panels respectively.

- Properties:

1) Dock

Allows you to attach Splitcontainer's border to its container. When you drop the splitcontainer on a form a Default value of Dock property is 'FILL' that means splitcontainer gets the whole area of form. In this example, I have set it to 'None' for the sack of demonstrating splitcontainer features.

2) Panel1Collapsed / Panel2Collapsed

Used to set or get whether the panel1 or Panel2 is collapsed or expanded. If you assign true to Panel1Collapsed then left or top panel will be collapsed and if you assign false to Panel1Collapsed then left or top panel will be expanded same behavior will be occurred with Panel2Collapsed property.

3) Orientation

Used to set or get the orientation of splitcontainer panel

4) Splitter Width

Splitter is a horizontal or vertical bar which use to set panel size, default size of splitter is 4 pixel but you can set or get the splitter width value programmatically by manipulating splitterwidth property.

Tab Control

- A TabControl is a collection of tab pages and a tab page is the actual control that hosts other child controls. TabPage class represents a tab page.
- Properties:

Property	Description
Alignment	Area of the control where the tabs are aligned.
Appearance	Visual appearance of the control's tabs.
DrawMode	A way that the control's tab pages are drawn.
HotTrack	Value indicating whether the control's tabs change in appearance when the mouse passes over them.
ImageList	The images to display on the control's tabs.
ItemSize	Size of the control's tabs.
Multiline	A value indicating whether more than one row of tabs can be displayed.
Padding	Amount of space around each item on the control's tab pages.
RowCount	Returns the number of rows that are currently being displayed in the control's tab strip.
SelectedIndex	The index of the currently-selected tab page.
SelectedTab	Currently selected tab page.
ShowToolTips	The value indicating whether a tab's ToolTip is shown when the mouse passes over the tab.
SizeMode	The way that the control's tabs are sized.
TabCount	Number of tabs in the tab strip.
TabPage	Returns the collection of tab pages in this tab control.

MenuStrip Control

- The **MenuStrip** control represents the container for the menu structure.
- The MenuStrip control works as the top-level container for the menu structure. The ToolStripMenuItem class and the ToolStripDropDownMenu class provide the functionalities to create menu items, sub menus and drop-down menus.
- Properties:

Sr.No.	Property & Description
1	CanOverflow Gets or sets a value indicating whether the MenuStrip supports overflow functionality.
2	GripStyle Gets or sets the visibility of the grip used to reposition the control.
3	MdiWindowListItem Gets or sets the ToolStripMenuItem that is used to display a list of Multiple-document interface (MDI) child forms.
4	ShowItemToolTips Gets or sets a value indicating whether ToolTips are shown for the MenuStrip.
5	Stretch Gets or sets a value indicating whether the MenuStrip stretches from end to end in its container.