1. **ListView control:**
   * The ListView control is used to display a list of items. Along with the TreeView control, it allows you to create a Windows Explorer like interface.
   * The ListView control displays a list of items along with icons. The Item property of the ListView control allows you to add and remove items from it.
   * The SelectedItem property contains a collection of the selected items.
   * The MultiSelect property allows you to set select more than one item in the list view. TheCheckBoxes property allows you to set check boxes next to the items.

* **Properties of the ListView Control:**
* The following are some of the commonly used properties of the ListView control:

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Property** | **Description** |
|  |  |  |
| 1 | **Alignment** | Gets or sets the alignment of items in the control. |
|  |  |  |
| 2 | **AutoArrange** | Gets or sets whether icons are automatically kept arranged. |
|  |  |  |
| 3 | **BackColor** | Gets or sets the background color. |
|  |  |  |
| 4 | **CheckBoxes** | Gets or sets a value indicating whether a check box |
|  |  | appears next to each item in the control. |
|  |  |  |
| 5 | **CheckedIndices** | Gets the indexes of the currently checked items in the |
|  |  | control. |
|  |  |  |
| 6 | **CheckedItems** | Gets the currently checked items in the control. |
|  |  |  |
| 7 | **Columns** | Gets the collection of all column headers that appear in the |
|  |  | control. |
|  |  |  |
| 8 | **GridLines** | Gets or sets a value indicating whether grid lines appear |
|  |  | between the rows and columns containing the items and |
|  |  | subitems in the control. |
|  |  |  |
| 9 | **HeaderStyle** | Gets or sets the column header style. |
|  |  |  |
| 10 | **HideSelection** | Gets or sets a value indicating whether the selected item in |
|  |  | the control remains highlighted when the control loses |
|  |  | focus. |
|  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  |  |  | | |  |  |
|  |  | | 11 | **HotTracking** | Gets or sets a value indicating whether the text of an item | | |  |  |
|  |  | |  |  | or subitem has the appearance of a hyperlink when the | | |  |  |
|  |  | |  |  | mouse pointer passes over it. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 12 | **HoverSelection** | Gets or sets a value indicating whether an item is | | |  |  |
|  |  | |  |  | automatically selected when the mouse pointer remains | | |  |  |
|  |  | |  |  | over the item for a few seconds. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 13 | **InsertionMark** | Gets an object used to indicate the expected drop location | | |  |  |
|  |  | |  |  | when an item is dragged within a ListView control. | |  |  |  |
|  |  | |  |  |  | |  |  |  |
|  |  | | 14 | **Items** | Gets a collection containing all items in the control. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 15 | **LabelWrap** | Gets or sets a value indicating whether item labels wrap | | |  |  |
|  |  | |  |  | when items are displayed in the control as icons. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 16 | **LargeImageList** | Gets or sets the ImageList to use when displaying items as | | |  |  |
|  |  | |  |  | large icons in the control. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 17 | **MultiSelect** | Gets or sets a value indicating whether multiple items can | | |  |  |
|  |  | |  |  | be selected. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 18 | **RightToLeftLayout** | Gets or sets a value indicating whether the control is laid | | |  |  |
|  |  | |  |  | out from right to left. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 19 | **Scrollable** | Gets or sets a value indicating whether a scroll bar is | | |  |  |
|  |  | |  |  | added to the control when there is not enough room to | | |  |  |
|  |  | |  |  | display all items. | |  |  |  |
|  |  | |  |  |  | |  |  |  |
|  |  | | 20 | **SelectedIndices** | Gets the indexes of the selected items in the control. | |  |  |  |
|  |  | |  |  |  | |  |  |  |
|  |  | | 21 | **SelectedItems** | Gets the items that are selected in the control. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 22 | **ShowGroups** | Gets or sets a value indicating whether items are displayed | | |  |  |
|  |  | |  |  | in groups. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 23 | **ShowItemToolTips** | Gets or sets a value indicating whether ToolTips are | | |  |  |
|  |  | |  |  | shown for the ListViewItem objects contained in | | |  |  |
|  |  | |  |  | theListView. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 24 | **SmallImageList** | Gets or sets the ImageList to use when displaying items as | | |  |  |
|  |  | |  |  | small icons in the control. | |  |  |  |
|  |  | |  |  |  | |  |  |  |
|  |  | | 25 | **Sorting** | Gets or sets the sort order for items in the control. | |  |  |  |
|  |  | |  |  |  | | |  |  |
|  |  | | 26 | **StateImageList** | Gets or sets the ImageList associated with application- | | |  |  |
|  |  | |  |  | defined states in the control. | |  |  |  |
|  |  | |  |  |  |  |  |  |  |
|  | |  |  | | |
| 27 | | **TopItem** | Gets or sets the first visible item in the control. | | |
|  | |  |  | | |
| 28 | | **View** | Gets or sets how items are displayed in the control. This | | |
|  | |  | property has the following values: | | |
|  | |  | LargeIcon - displays large items with a large 32 x | | |
|  | |  | 32 pixels icon. | | |
|  | |  | SmallIcon - displays items with a small 16 x 16 | | |
|  | |  | pixels icon | | |
|  | |  | List - displays small icons always in one column | | |
|  | |  | Details - displays items in multiple columns with | | |
|  | |  | column headers and fields | | |
|  | |  | Tile - displays items as full-size icons with item | | |
|  | |  | labels and sub-item information. | | |
|  | |  |  | | |
| 29 | | **VirtualListSize** | Gets or sets the number of ListViewItem objects contained | | |
|  | |  | in the list when in virtual mode. | | |
|  | |  |  | | |
| 30 | | **VirtualMode** | Gets or sets a value indicating whether you have provided | | |
|  | |  | your own data-management operations for the ListView | | |
|  | |  | control. | | |
|  | |  |  | | |

* Methods of the ListView Control:
* The following are some of the commonly used methods of the ListView control:

**Sr.No.** **Method Name & Description**

1. **Clear**

Removes all items from the ListView control.

* 1. **ToString**

Returns a string containing the string representation of the control.

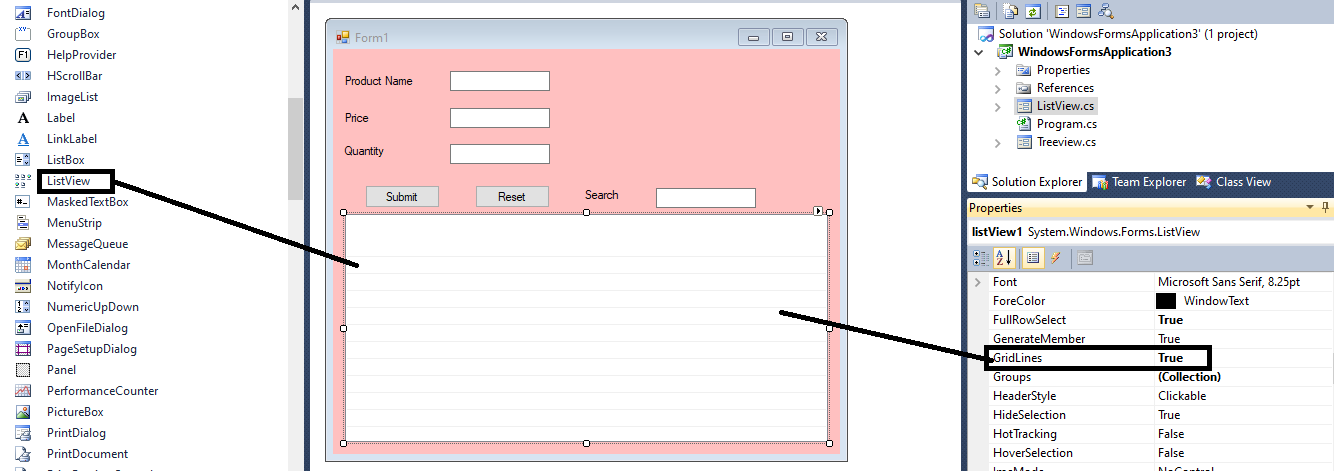
* Events of the ListView Control:
* The following are some of the commonly used events of the ListView control:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Sr.No.** | **Event** | **Description** |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  | 1 | **ColumnClick** | Occurs when a column header is clicked. |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 2 | **ItemCheck** | Occurs when an item in the control is checked or |
|  |  | unchecked. |
|  |  |  |
| 3 | **SelectedIndexChanged** | Occurs when the selected index is changed. |
|  |  |  |
| 4 | **TextChanged** | Occurs when the Text property is changed. |
|  |  |  |

// Example:

Design a window form as given below screenshot:



Step1:

Write down the below code on form load event for adding columns in List View control.

private void Form1\_Load(object sender, EventArgs e)

{

listView1.View = View.Details;

listView1.Columns.Add("Product Name");

listView1.Columns.Add("Price");

listView1.Columns.Add("Quantity");

}

Step2:

Write down the below code on Submit button click event for inserting values in List View control.

private void submit\_Click(object sender, EventArgs e)

{

ListViewItem lvwItem = listView1.Items.Add(textBox1.Text);

lvwItem.SubItems.Add(textBox2.Text);

lvwItem.SubItems.Add(textBox3.Text);

}

Code for removing items from the list view while clicking on any list view item:

private void listView1\_MouseClick(object sender, MouseEventArgs e)

{

textBox1.Text = listView1.SelectedItems[0].SubItems[0].Text;

textBox2.Text = listView1.SelectedItems[0].SubItems[1].Text;

textBox3.Text = listView1.SelectedItems[0].SubItems[2].Text;

foreach (ListViewItem eachItem in listView1.SelectedItems)

{

listView1.Items.Remove(eachItem);

}

}

Code for searching items from the list view while write any text in search textbox:

private void searchBox\_TextChanged(object sender, EventArgs e)

{

for (int i = listView1.Items.Count - 1; i >= 0; i--)

{

var item = listView1.Items[i];

if (item.Text.ToLower().Contains(searchBox.Text.ToLower()) )

{

item.BackColor = SystemColors.Highlight;

item.ForeColor = SystemColors.HighlightText;

}

else

{

listView1.Items.Remove(item);

}

if (listView1.SelectedItems.Count == 1)

{

listView1.Focus();

}

}

}