**The laboratory work 14**

**Creating Application with SfListView control**

In this walk through, user will create WinForms application that contains SfListView control.

**Creating the Project**

Create a new Windows Forms project in Visual Studio to display the SfListView with data objects.

**Adding Control via Designer**

The SfListView control can be added to the application by dragging it from the toolbox and dropping it in designer. The required assembly references will be added automatically.



**Adding Control in Code**

To add control manually, follow the steps:

1. Add the following required assembly references to the project:
	* Syncfusion.Core.WinForms
	* Syncfusion.DataSource.WinForms
	* Syncfusion.GridCommon.WinForms
	* Syncfusion.SfListView.WinForms
2. Create the SfListView control instance and add it to the form.
* C#
* VB.NET

using Syncfusion.WinForms.ListView;

 namespace WindowsFormsApplication1

 {

 public partial class Form1 : Form

 {

 public Form1()

 {

 InitializeComponent();

 SfListView sfListView1 = new SfListView ();

 sfListView1.Location = new Point(100, 100);

 sfListView1.Size = new Size(300,320);

 this.Controls.Add(sfListView1);

 }

 }

 }

**Creating data for sample application**

To create the data for sample application, follow the steps:

1. Create a data object class, name it as “CountryInfo” and declare the properties.
	* C#
	* VB.NET

 public class CountryInfo

{

 public string CountryName { get; set; }

 public string Continent { get; set; }

 }

 Create a List collection initialized in GetDataSource method to add several data objects.

* C#
* VB.NET
1. public List<CountryInfo> GetDataSource()
2. {
3. List<CountryInfo> countryInfoCollection = new List<CountryInfo>();
4. countryInfoCollection.Add(new CountryInfo() { CountryName = "China", Continent = "Asia" });
5. countryInfoCollection.Add(new CountryInfo() { CountryName = "India", Continent = "Asia" });
6. countryInfoCollection.Add(new CountryInfo() { CountryName = "Japan", Continent = "Asia" });
7. countryInfoCollection.Add(new CountryInfo() { CountryName = "Malaysia", Continent = "Asia" });
8. countryInfoCollection.Add(new CountryInfo() { CountryName = "Singapore", Continent = "Asia" });
9. countryInfoCollection.Add(new CountryInfo() { CountryName = "Kenya", Continent = "Africa" });
10. countryInfoCollection.Add(new CountryInfo() { CountryName = "Nigeria", Continent = "Africa" });
11. countryInfoCollection.Add(new CountryInfo() { CountryName = "South Africa", Continent = "Africa" });
12. countryInfoCollection.Add(new CountryInfo() { CountryName = "Uganda", Continent = "Africa" });
13. countryInfoCollection.Add(new CountryInfo() { CountryName = "Zimbabwe", Continent = "Africa" });
14. countryInfoCollection.Add(new CountryInfo() { CountryName = "France", Continent = "Europe" });
15. countryInfoCollection.Add(new CountryInfo() { CountryName = "Germany", Continent = "Europe" });
16. countryInfoCollection.Add(new CountryInfo() { CountryName = "Italy", Continent = "Europe" });
17. countryInfoCollection.Add(new CountryInfo() { CountryName = "Spain", Continent = "Europe" });
18. countryInfoCollection.Add(new CountryInfo() { CountryName = "United Kingdom", Continent = "Europe" });
19. countryInfoCollection.Add(new CountryInfo() { CountryName = "Canada", Continent = "North America" });
20. countryInfoCollection.Add(new CountryInfo() { CountryName = "Cuba", Continent = "North America" });
21. countryInfoCollection.Add(new CountryInfo() { CountryName = "Jamaica", Continent = "North America" });
22. countryInfoCollection.Add(new CountryInfo() { CountryName = "Mexico", Continent = "North America" });
23. countryInfoCollection.Add(new CountryInfo() { CountryName = "United States of America", Continent = "North America" });
24. countryInfoCollection.Add(new CountryInfo() { CountryName = "Australia", Continent = "Oceania" });
25. countryInfoCollection.Add(new CountryInfo() { CountryName = "New Zealand", Continent = "Oceania" });
26. countryInfoCollection.Add(new CountryInfo() { CountryName = "Argentina", Continent = "South America" });
27. countryInfoCollection.Add(new CountryInfo() { CountryName = "Brazil", Continent = "South America" });
28. countryInfoCollection.Add(new CountryInfo() { CountryName = "Chile", Continent = "South America" });
29. countryInfoCollection.Add(new CountryInfo() { CountryName = "Colombia", Continent = "South America" });
30. countryInfoCollection.Add(new CountryInfo() { CountryName = "Uruguay", Continent = "South America" });
31. return countryInfoCollection;
32. }

**Binding to data**

To bind the SfListView to data, set the [SfListView.DataSource](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_DataSource) property to an IEnumerable implementation.
You can bind a property of the underlying data source to display the SfListView by using the [DisplayMember](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_DisplayMember) property.

* C#
* VB.NET

sfListView1.DataSource = GetDataSource();

sfListView1.DisplayMember = "CountryName";



**Grouping**

The Windows Forms ListView (SfListView) allows displaying the items in a group by using the [SfListView.View.GroupDescriptors](https://help.syncfusion.com/cr/windowsforms/Syncfusion.DataSource.DataSource.html#Syncfusion_DataSource_DataSource_GroupDescriptors) property. Create a [GroupDescriptor](https://help.syncfusion.com/cr/windowsforms/Syncfusion.DataSource.GroupDescriptor.html) for the property to be grouped and add it in the View.GroupDescriptors collection.

GroupDescriptor object holds the following properties:

• `PropertyName`: Describes name of the property to be grouped.

• `KeySelector`: Describes selector to return the group key.

• `Comparer`: Describes comparer to be applied when sorting takes place.

* C#
* VB.NET

listView.View.GroupDescriptors.Add(new GroupDescriptor()

 {

 PropertyName = "Continent",

 });



**Sorting**

The SfListView allows sorting on its data by using the [SfListView.View.SortDescriptors](https://help.syncfusion.com/cr/windowsforms/Syncfusion.DataSource.DataSource.html#Syncfusion_DataSource_DataSource_SortDescriptors) property. Create a [SortDescriptor](https://help.syncfusion.com/cr/windowsforms/Syncfusion.DataSource.SortDescriptor.html) for the property to be sorted and add it into the View.SortDescriptors collection.

SortDescriptor object holds the following three properties:

• `PropertyName`: Describes name of the sorted property.

• `Direction`: Describes an object of type `ListSortDirection` defines the sorting direction.

• `Comparer`: Describes a comparer to be applied when sorting takes place.

* C#
* VB.NET

listView.View.SortDescriptors.Add(new SortDescriptor()

{

 PropertyName = "Continent",

 Direction = ListSortDirection.Descending,

});



**Filtering**

The SfListView support to filter the records in view by setting predicate to the [SfListView.View.Filter](https://help.syncfusion.com/cr/windowsforms/Syncfusion.DataSource.DataSource.html#Syncfusion_DataSource_DataSource_Filter) property. Call the [View.RefreshFilter](https://help.syncfusion.com/cr/windowsforms/Syncfusion.DataSource.DataSource.html#Syncfusion_DataSource_DataSource_RefreshFilter) method after assigning the Filter property for refreshing the view.
To filter the items based on the Continent property of the underlying data, follow the code example.

* C#
* VB.NET

listView.View.Filter = CustomFilter;

listView.View.RefreshFilter();

public bool CustomFilter(object obj)

{

 if ((obj as Country).Continent == "Asia" || (obj as Country).Continent == "North America" || (obj as Country).Continent == "Oceania")

 return true;

 return false;

}



**Selection**

SfListView selects an item by setting the [SfListView.SelectionMode](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_SelectionMode) property to One, MultiSimple, MultiExtended, and None based on the requirements. Selected item information can be tracked by using the [SfListView.SelectedItem](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_SelectedItem), [SfListView.SelectedIndex](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_SelectedIndex), and [SfListView.SelectedItems](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_SelectedItems) properties.

The selection operations can be handled with the help of [SelectionChanging](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_SelectionChanging) and [SelectionChanged](https://help.syncfusion.com/cr/windowsforms/Syncfusion.WinForms.ListView.SfListView.html#Syncfusion_WinForms_ListView_SfListView_SelectionChanged) events of the SfListView.

* C#
* VB.NET

listView.SelectionMode = SelectionMode.One;

