User control – advanced and good practice of using C# user control

In Windows forms applications.

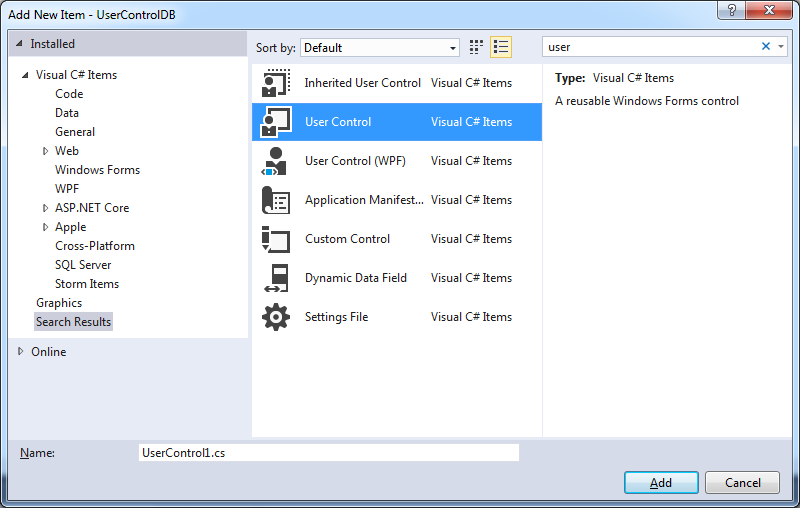
<https://github.com/EliArad/C-UserControl>

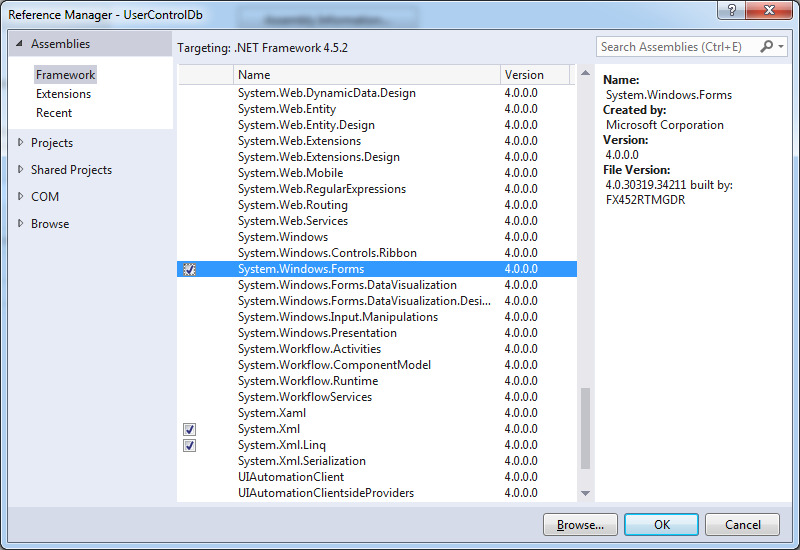
We will learn how to use the Windows c# user control in a way that it works like a repeater from data base or a list.

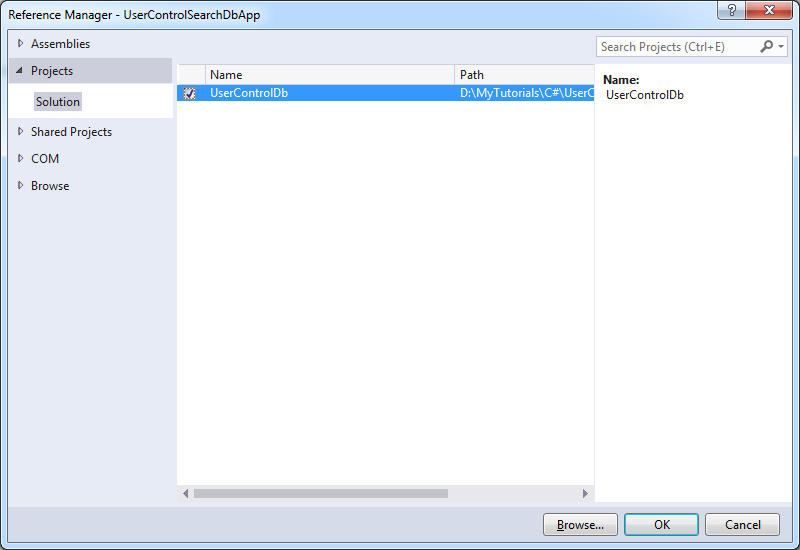
And show how we can search and dynamically show the results

First, let’s create two projects

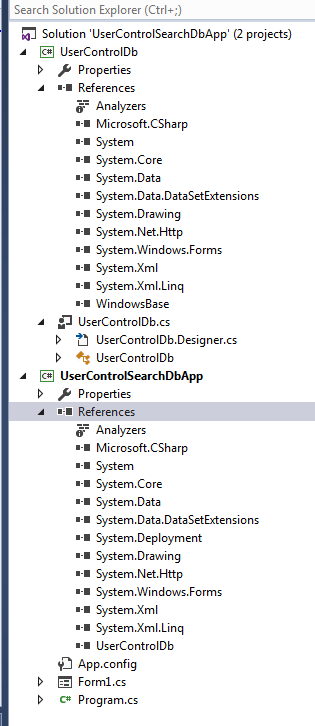
1. Windows form application that will host our user control
2. Instead of creating the user control inside our windows form, lets add it as a separate class library.
3. I will show the steps on the video







Our initial project should be as follow:



We have two project app and class lib dll.

The dll have the user control with assembly of windows forms.

This will help us to add the user control easily to different applications.

The app , windows form app, hold a reference to the class library.

Lets create a student class

public class Student

{

public string firstName { get; set; }

public string lastName { get; set; }

public int age { get; set; }

public bool male { get; set; }

public bool likeSports { get; set; }

}

And add name in a list

In the windows form

void InitDb()

{

Student n = new Student

{

firstName = "Eli",

lastName = "Arad",

age = 40,

likeSports = true,

male = true

};

m\_students.Add(n);

n = new Student

{

firstName = "Yo",

lastName = "Bo",

age = 30,

likeSports = true,

male = true

};

m\_students.Add(n);

n = new Student

{

firstName = "Ami",

lastName = "Rabon",

age = 30,

likeSports = true,

male = true

};

m\_students.Add(n);

n = new Student

{

firstName = "Joe",

lastName = "Stample",

age = 19,

likeSports = true,

male = true

};

m\_students.Add(n);

n = new Student

{

firstName = "David",

lastName = "Tony",

age = 52,

likeSports = true,

male = true

};

m\_students.Add(n);

n = new Student

{

firstName = "Yariv",

lastName = "Logasi",

age = 52,

likeSports = true,

male = true

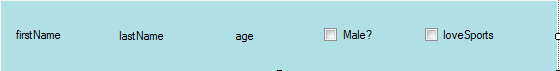
};

m\_students.Add(n);

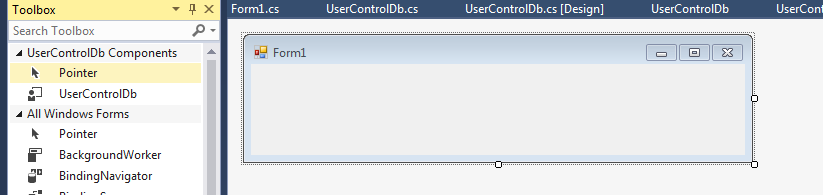
}

What we want to do now , is to show it in the user control as a list.

I create the following user control

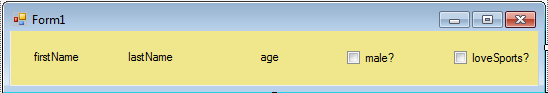


Add it to the windows form:



From here:

We put one on the form



Just because we want to take the initialization from visual studio:

F12 on InitializeComponent

And grab those lines:

this.userControlDb1 = new UserControlDb.UserControlDb();

this.userControlDb1.BackColor = System.Drawing.Color.Khaki;

this.userControlDb1.Location = new System.Drawing.Point(1, -1);

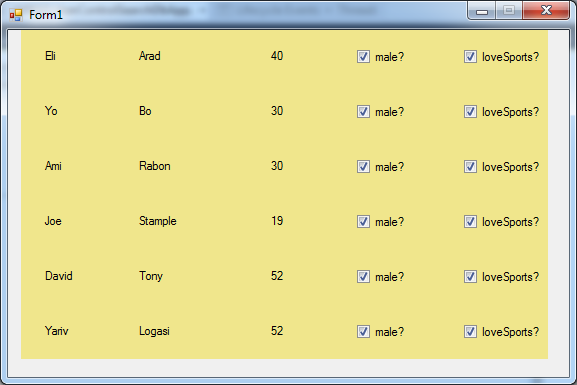
this.userControlDb1.Name = "userControlDb1";

this.userControlDb1.Size = new System.Drawing.Size(527, 55);

this.userControlDb1.TabIndex = 0;

Delete and add a panel:

To create the list:



I wrote that code:

void AddStudents()

{

int y = 0;

int i = 0;

foreach (Student s in m\_students)

{

UserControlDb u = new UserControlDb();

u.BackColor = System.Drawing.Color.Khaki;

u.Location = new System.Drawing.Point(1, -1 + y);

u.Name = "userControlDb " + u;

u.Size = new System.Drawing.Size(527, 55);

u.TabIndex = i++;

u.AddStudent(s);

panel1.Controls.Add(u);

y += u.Height;

}

panel1.Height = y;

this.Height = y + 55;

}

namespace UserControlDbLib

{

public partial class UserControlDb : UserControl

{

public UserControlDb()

{

InitializeComponent();

}

public void AddStudent(Student s)

{

lblAge.Text = s.age.ToString();

lblFirstName.Text = s.firstName;

lblLastName.Text = s.lastName;

chkMale.Checked = s.male;

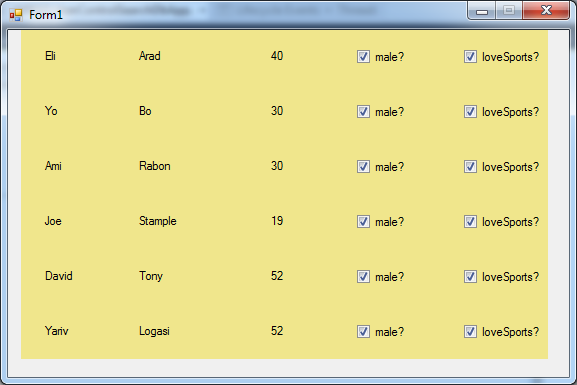
chkLoveSports.Checked = s.likeSports;

}

}

}

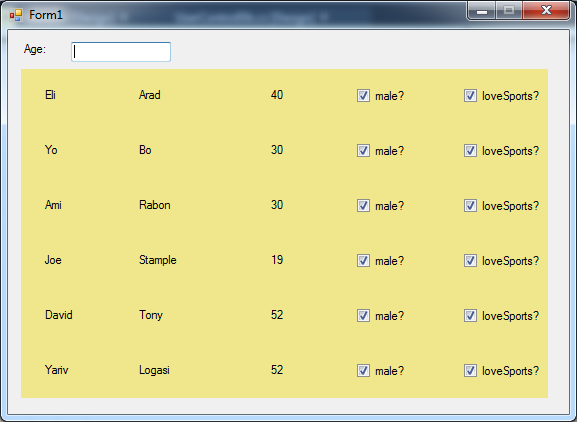
Now we have our list ready



We can limit the view and add search to dynamically show results.

It is linq sql or entity framework, we will display the results that we want , by clearing the panel and add the results on the fly.

Add the following textbox:



And the logic on the code behind:

private void textBox1\_TextChanged(object sender, EventArgs e)

{

int ageToSearch;

bool b = int.TryParse(textBox1.Text, out ageToSearch);

if (b)

{

var filteredOrders = (from s in m\_students

where s.age > ageToSearch

select s).ToList();

panel1.Controls.Clear();

AddStudents(filteredOrders);

} else

{

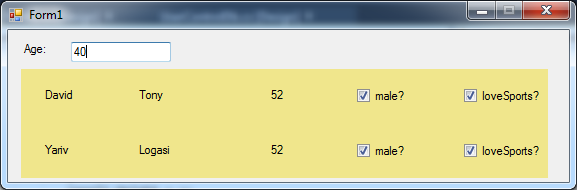
AddStudents(m\_students);

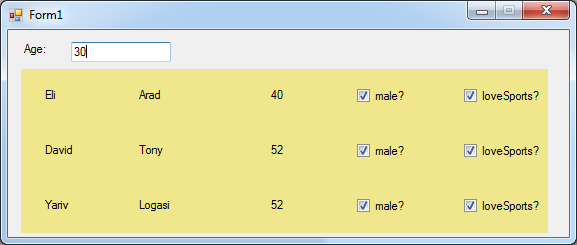
}

}

Results:

Search will show that:





That way we can search by more fields

And the data can be handle from data base and not from static list as we did here.