

Forum:[Hack Forums](#)

Thread: [c# listview search/filter without datatable or binding](#)

wanbetaaler - October 18, 2017, 7:00 am

Hello,

I do not know for sure if this is the right section as i can not find a c# thread section and a search didnt help much as i need 3 letters and c# contains 2 characters haha. Now to the problem.

In C# i have a listview which i manually add items to and i add it to a list<class>.

I want to know if it is possible to do a search in the c# listview with a like, so if i type something like; bos i want to display different listview rows in the listview that contain something with bos so it can be in the middle ending of starting of a word. Also i want to know if it is possible to search on all the collumns instead of just the first one?

For visual representation, I have 2 forms, one where I enter the details to be added to the list and a main form where i show all the passwords and its different features. Check for screenshots below.

Spoiler [\(Click to View\)](#)

The architecture i use is the following,

I have a precreated listview in the main form. In the public constructor of the main form i create an instance of the new form which i give the argument of the existing listview with it.

The way i get the listview as an argument of the second form

Code public AddPassword(ListView lvpasswords)

```
{
    InitializeComponent();
    lv = lvpasswords;
}
```

The way i set the listview in the first and main form.

Code public PassKeeper()

```
{
    InitializeComponent();
    adpw = new AddPassword(lvPasswords);
}
```

My code for adding to the listview will be shown below.

Here i show the second form for adding a new password.

and add it to the list<class> of the data using the class i created and filled in the second form.

Spoiler [\(Click to View\)](#)

Code private void btnAddPassword_Click(object sender, EventArgs e)

```
{
    adpw.ShowDialog();
    datapwList.Add(adpw.datapwClass);
    updateCounter(); // this is to update the count of saved passwords
}
```

I created a construction of the class as a public class in the second form like this

Code public Data datapwClass; Adding the data to the listview.

Spoiler ([Click to View](#))

Code private void btnSavePassword_Click(object sender, EventArgs e)

```
{
    if (!string.IsNullOrEmpty(TbDomain.Text) &&
        !string.IsNullOrEmpty(TbUsername.Text) &&
        !string.IsNullOrEmpty(TbPassword.Text))
    {
        string domain = TbDomain.Text;
        string username = TbUsername.Text;
        string password = TbPassword.Text;

        ListViewItem items = new ListViewItem(domain);
        items.SubItems.Add(username);
        items.SubItems.Add(password);

        lv.Items.Add(items);

        datapwClass = new Data();
        datapwClass.domain = domain;
        datapwClass.username = username;
        datapwClass.password = password;

        clearFields();
        Close();
    }
    else
    {
        MessageBox.Show("Please fill in all the fields");
    }
}
```

The class looks like this

Spoiler ([Click to View](#))

Zens - October 18, 2017, 7:42 am

Wrong section this belong to C# section, the sub forums of VB.net

wanbetaaler - October 18, 2017, 7:54 am

(Oct 18, 2017 07:42 AM)Zens Wrote: [\[CITE\]](#) Wrong section this belong to C# section, the sub forums of VB.net

Thank you bro, i will close this thread and copy my thread over there.

{edit}

Oh did you move this for me?

Thank you, could you remove the thread i created in the vb.net main section and tell me if i can move my threads as well and how i could perform such an action.

Zens - October 18, 2017, 8:00 am

(Oct 18, 2017 07:54 AM)wanbetaaler Wrote: [\[CITE\]](#) Thank you bro, i will close this thread and copy my thread over there.

{edit}

Oh did you move this for me?

Thank you, could you remove the thread i created in the vb.net main section and tell me if i can move my threads as well and how i could perform such an action.

No, i'm not a staff i can't move, edit, etc.

Since you are l33t you can close this thread. :)

Don't cross post tho. I believe staff will move this to the correct section so you not have to open a new thread ?

Ninja - October 18, 2017, 8:14 am

Why without databinding though?

rizkyaqew - October 18, 2017, 8:17 am

If you think you need to filter the displayed items, you can use DataGridView that binds to BindingSource class, where each item you add to BindingSource will be reflected to DataGridView. However, if you want to filter the rows through list view, I'll give you some example ? :

```
Code // Query text through list items and sub items
// collect the indexes of each row.
static int[] FindText ( ListView list , string text )
{
    // Use OrdinalIgnoreCase to perform case-insensitive comparison
    // Use Ordinal to perform case-sensitive comparison
    StringComparer comparer = StringComparer.OrdinalIgnoreCase;
    List<int> indices = new List<int>();
    for ( int n = 0 ; n < list.Items.Count ; ++n )
    {
        ListViewItem item = list.Items [ n ];
        if ( comparer.Equals ( item.Text , text ) ) { indices.Add(n); break;}
        for ( int i = 0 ; i < item.SubItems.Count ; ++i )
            if ( comparer.Equals ( text, item.SubItems [ i ].Text ) ) {indices.Add(n); break;} else
continue;
    }
    return indices.ToArray();
}

// copy founded rows from list source to list display
static void DisplayOnlyRows(ListView listSource, ListView listDisplay, string text)
{
    int[] indices = FindText(listSource, text);
    for(int n=0; n < indices.Length; listDisplay.Items.Add(listSource.Items[n]), ++n);
}

void Button1_Click(object sender, EventArgs e)
{
    string textToFind = textBox1.Text; // get the text to find
    this.listDisplay.Items.Clear(); // clear the display items
```

```

        DisplayOnlyRows(this.listSource, this.listDisplay, textToFind); // copy the rows to display
    }
}

```

wanbetaaler - October 18, 2017, 8:48 am

(Oct 18, 2017 08:14 AM)Ninja Wrote: [\[CITE\]](#) Why without databinding though?
To be honest, its because i never used databinding and it feares me xD.

(Oct 18, 2017 08:17 AM)rizkyaqew Wrote: [\[CITE\]](#) If you think you need to filter the displayed items, you can use DataGridView that binded to BindingSource class, where each items you add to BindingSource will reflected to DataGridView. However, if you want to filter the rows through list view, I'll give you some example ? :

Code // Query text through list items and sub items

// collect the indexes of each rows.

static int[] FindText (ListView list , string text)

{

// Use OrdinalIgnoreCase to perform case-insensitive comparison

// Use Ordinal to perform case-sensitive comparison

StringComparer comparer = StringComparer.OrdinalIgnoreCase;

List<int> indices = new List<int>();

for (int n = 0 ; n < list.Items.Count ; ++n)

{

ListViewItem item = list.Items [n];

if (comparer.Equals (item.Text , text)) { indices.Add(n); break;}

for (int i = 0 ; i < item.SubItems.Count ; ++i)

if (comparer.Equals (text, item.SubItems [i].Text)) {indices.Add(n); break;} else

continue;

}

return indices.ToArray();

}

// copy founded rows from list source to list display

static void DisplayOnlyRows(ListView listSource, ListView listDisplay, string text)

{

int[] indices = FindText(listSource, text);

for(int n=0; n < indices.Length; listDisplay.Items.Add(listSource.Items[n]), ++n);

}

void Button1_Click(object sender, EventArgs e)

{

string textToFind = textBox1.Txt; // get the text to find

this.listDisplay.Items.Clear(); // clear the display items

DisplayOnlyRows(this.listSource, this.listDisplay, textToFind); // copy the rows to display

}

}

Dear rizkyaqew,

I have tryed your solution, first of all i dont get the solution and second of all I could net get this solution to work. Any further help would be appriciated. Inbox me your contact details?
and is it possible to compare 2 strings that are like each other instead of identical to each other?

rizkyaqew - October 18, 2017, 10:20 am

(Oct 18, 2017 08:48 AM)wanbetaaler Wrote: [\[CITE\]](#) To be honest, its because i never used databinding and it feares me xD.

Dear rizkyaqew,

I have tryed your solution, first of all i dont get the solution and second of all I could net get this solution to work. Any further help would be appreciated. Inbox me your contact details?
and is it possible to compare 2 strings that are like each other instead of identical to each other?
To compare two strings that like each other ? Ok, I'll give your more contrast code in following examples :

```
Code public bool MaybeMatch(string source, string pattern)
```

```
{
    // If you want to perform case-sensitive comparision
    // change the OrdinalIgnoreCase with Ordinal
    StringComparison mode = StringComparison.OrdinalIgnoreCase;

    // Will returns true if the source same with pattern,
    // the source starts with pattern, the source ends with pattern,
    // or at least the source is contains the pattern in the character
    // sequences. Otherwise, will returns false;
    return string.Equals(source, pattern, mode) ||
        source.StartsWith(pattern, mode) || source.EndsWith(pattern, mode) ||
        source.IndexOf(pattern, mode) >= 0;
}
```

```
// usages
```

```
public void FindMatchedRows(ListView yourListView, ListView listToDisplayMatchedItems, string
requestedText)
```

```
{
    // create a new list to store the matched row indexes.
    List<int> map = new List<int>();
    // here you will loops all items in list view to search
    for(int n = 0; n < yourListView.Items.Count; ++n)
    {
        // first you check the item text, if match, then add to list
        if(MaybeMatch(yourListView.Items[n].Text, requestedText))
        {
            // We've found matched requested text pattern, add to list
            map.Add(n);
            // Since we have found the text, just skip the sub item comparisons
            continue;
        }
        // Now, we will compare the sub items displayed text,
        // since the first column text is not match.
        bool found = false;
        for(int i = 0; !found && i < yourListView.Items[n].SubItems.Count; ++i)
        {
            // just set the 'found' variable, if true, the nested loop will
            // automatically break.
            found = MaybeMatch(yourListView.Items[n].SubItems[i].Text, requestedText);
        }
    }
}
```

```

        // If the sub items has contains the requested text, add it to list
        if(found) map.Add(n);
    }
    // check whether the result is present
    if(map.Count > 0)
    {
        // now you get the rows index that have matched strings.
        // what will you do now? You can do it here ..
        // the following code is displaying the founded items in listview
        // to another list view

        // I need to clear the output listview first.
        listToDisplayMatchedItems.Items.Clear();
        // Loop through indexes map collection
        for(int n=0; n < map.Count; n++)
        {
            // here is the item with matched strings
            ListViewItem item = yourListView.Items[map[n]];
            // add the item to another list view
            listToDisplayMatchedItems.Items.Add(item);
        }
        // Now the list view that used to display matched strings will
        // display the matched rows.
        listToDisplayMatchedItems.Refresh();
        MessageBox.Show(string.Format("We've found {0} rows that identical to \"{1}\", thanks for
your attention.", map.Count, requestedText),
        "Success Filters", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        // this block is to handle when the search is found nothing.
        MessageBox.Show(string.Format("The \"{0}\" is not match with anything data in list, sorry.",
requestedText), "Filtering Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

The above code is just using basic .NET string comparison, for further comparison, like pattern search "t*.txt", use Regular Expression, the .NET class that provides regular expression is System.Text.RegularExpressions.Regex, there should be plenty examples around MSDN sites. ?

If you found error due miss-typing or compiler errors, forgive me, since I dont open my Visual Studio when replying your post.

You can feel free ask me if you still got the troubles. ?

wanbetaaler - October 18, 2017, 10:50 am

(Oct 18, 2017 10:20 AM)rizkyaqew Wrote: [CITE](#) To compare two strings that like each other ? Ok, I'll give you more contrast code in following examples :

Spoiler [Click to View](#)

Code public bool MaybeMatch(string source, string pattern)

```

{
    // If you want to perform case-sensitive comparison
    // change the OrdinalIgnoreCase with Ordinal
    StringComparison mode = StringComparison.Ordinal;
}

```

```

// Will returns true if the source same with pattern,
// the source starts with pattern, the source ends with pattern,
// or at least the source is contains the pattern in the character
// sequences. Otherwise, will returns false;
return string.Equals(source, pattern, mode) ||
source.StartsWith(pattern, mode) || source.EndsWith(pattern, mode) ||
source.IndexOf(pattern, mode) >= 0;
}

// usages
public void FindMatchedRows(ListView yourListView, ListView listToDisplayMatchedItems, string
requestedText)
{
    // create a new list to store the matched row indexes.
    List<int> map = new List<int>();
    // here you will loops all items in list view to search
    for(int n = 0; n < yourListView.Items.Count; ++n)
    {
        // first you check the item text, if match, then add to list
        if(MaybeMatch(yourListView.Items[n].Text, requestedText))
        {
            // We've found matched requested text pattern, add to list
            map.Add(n);
            // Since we have found the text, just skip the sub item comparisons
            continue;
        }
        // Now, we will compare the sub items displayed text,
        // since the first column text is not match.
        bool found = false;
        for(int i = 0; !found && i < yourListView.Items[n].SubItems.Count; ++i)
        {
            // just set the 'found' variable, if true, the nested loop will
            // automatically break.
            found = MaybeMatch(yourListView.Items[n].SubItems[i].Text, requestedText);
        }
        // If the sub items has contains the requested text, add it to list
        if(found) map.Add(n);
    }
    // check whether the result is present
    if(map.Count > 0)
    {
        // now you get the rows index that have matched strings.
        // what will you do now? You can do it here ..
        // the following code is displaying the founded items in listview
        // to another list view

        // I need to clear the output listview first.
        listToDisplayMatchedItems.Items.Clear();
        // Loop through indexes map collection
        for(int n=0; n < map.Count; n++)
        {
            // here is the item with matched strings
            ListViewItem item = yourListView.Items[map[n]];

```

```

        // add the item to another list view
        listToDisplayMatchedItems.Items.Add(item);
    }
    // Now the list view that used to display matched strings will
    // display the matched rows.
    listToDisplayMatchedItems.Refresh();
    MessageBox.Show(string.Format("We've found {0} rows that identical to \"{1}\", thanks for
your attention.", map.Count, requestedText),
    "Success Filters", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
    // this block is to handle when the search is found nothing.
    MessageBox.Show(string.Format("The \"{0}\" is not match with anything data in list, sorry.",
requestedText), "Filtering Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

The above code is just using basic .NET string comparison, for further comparison, like pattern search "t*.txt", use Regular Expression, the .NET class that provides regular expression is System.Text.RegularExpressions.Regex, there should be plenty examples around MSDN sites. ?

If you found error due miss-typing or compiler errors, forgive me, since I don't open my Visual Studio when replying your post.

You can feel free ask me if you still got the troubles. ?

Thank you very much, I have one more question tho.

When displaying the matching listview items i get the problem that it says i can not have the same item at multiple places, and that i need to remove it copy or clone it.

How can i copy/clone a listview item? could you explain that as well..

and after re-reading your first example code it was much more clearer to me after beautifying it a bit in vs.

I am reading the second post as we speak. But cloning/copying a listview item is always handy to know. Do you know how to do it?

rizkyaqew - October 18, 2017, 11:46 am

Both of ListViewItem or ListViewItem.ListViewSubItem classes doesn't supporting cloning by default, you need to create such as helper function to clone it, before it, you need to know that cloning is equivalent to creating a new same object with same variables. So you should create a function that re create a new items based from supplied items to copy ? :

```

Code public static ListViewItem CopyItem(ListViewItem source)
{
    ListViewItem item = new ListViewItem();
    item.Text = source.Text;
    // copy the other properties you need
    for(int n=0; n < source.SubItems.Count; ++n)
    {
        var sub = source.SubItems[n];
        var added = item.SubItems.Add(sub.Text);
        // do copy the sub items properties you need.
    }
    return item;
}

```

```
}
```

```
// if you just want to display the matched items on same list view  
// you can add the non matched items to temporary list, like :  
private List<ListViewItem> backup = new List<ListViewItem>();
```

```
void DisplayMatchedOnly(List<int> map)  
{  
    // clear the backup list first  
    this.backup.Clear();  
    // declare the list of indexes to remove from list view  
    List<int> toRemove = new List<int>();  
    // filter the items to hides.  
    for(int n=0; n < this>YourListView.Items.Count; ++n)  
    {  
        // keep the track of original index, so you can restore it later.  
        this>YourListView.Items[n].Tag = n;  
        if(map.IndexOf(n) < 0)  
        {  
            this.backup.Add(this>YourListView.Items[n]); // keep the item at backup storage  
            toRemove.Add(n); // add the index of this item to remove  
        }  
    }  
    // Now removes don't need to display  
    for(int n =0; n < toRemove.Count; ++n)  
    {  
        this>YourListView.Items.RemoveAt(toRemove[n]);  
    }  
}  
// here are the method to restore the removed items from backup.  
// Call this after you filter the items  
void RestoreAllItems()  
{  
    for(int n=0; n < this.backup.Count; this>YourListView.Items.Add(this.backup[n]), ++n);  
    this>YourListView.ListViewItemSorter = ItemComparer.Singleton;  
    this>YourListView.Sort();  
    this.backup.Clear();  
}
```

```
class ItemComparer : IComparer  
{  
    private ItemComparer(){}  
  
    public static readonly Singleton = new ItemComparer();  
    int Compare(object x, object y)  
    {  
        ListViewItem a =(ListViewItem)x, b = (ListViewItem)y;  
        return a.Tag is int && b.Tag is int ? ((int)a.Tag).CompareTo((int)b.Tag) :  
string.Compare(a.Text, b.Text);  
    }  
}
```

Rizka, thank you for your examples.

I would like to post my outcome in this thread so other people can see what my solution was.

Code //test population

```
string[,] rows = { { "piet", "34", "amsterdam" }, { "jan", "21", "maaslander" }, { "erik", "12",  
"Rotterdam" }, { "sjaak", "18", "rotterdam" } };  
ListViewItem[] itemsC;  
  
HashSet<int> indeces;  
string pattern;  
  
public Form1()  
{  
    InitializeComponent();  
    populateList();  
}  
  
private void textBox1_TextChanged(object sender, EventArgs e)  
{  
    TextBox tb = (TextBox)sender;  
    pattern = tb.Text;  
    indeces = new HashSet<int>();  
    if (pattern.Length > 0)  
    {  
        for (int i = 0; i < itemsC.Count(); i++)  
        {  
            ListViewItem CurrItem = itemsC[i];  
            // checking the results on the first second and third collumn of the listview.  
            if (maybeMatch(CurrItem.Text, pattern) || maybeMatch(CurrItem.SubItems[1].Text,  
pattern) || maybeMatch(CurrItem.SubItems[2].Text, pattern))  
            {  
                if (!indeces.Contains(i))  
                {  
                    indeces.Add(i);  
                }  
                Console.WriteLine("Index added: "+ i);  
                Console.WriteLine("count" + indeces.Count);  
            }  
        }  
    }  
  
    this.listSource.Items.Clear();  
    label1.Text = String.Format("{0} Matches found", indeces.Count.ToString()) ;  
    foreach (var index in indeces)  
    {  
        Console.WriteLine("The index: "+index+" is in the array");  
        this.listSource.Items.Add(itemsC[index]);  
    }  
}  
else  
{  
    displayAllItems();  
    indeces.Clear();  
    Console.WriteLine("Indces cleared!");  
}
```

```

        label1.Text = String.Format("{0} Matches found", indeces.Count.ToString());
    }
}

```

```

private bool maybeMatch(string source, string patternX)
{
    string pattern = Regex.Escape(patternX.ToLower());
    string sentence = source.ToLower();
    foreach (Match match in Regex.Matches(sentence, pattern))
    {
        return true;
    }
    return false;
}

```

```

void displayAllItems()
{
    this.listSource.Items.Clear();
    foreach (var item in itemsC)
    {
        listSource.Items.Add(item);
    }
}

```

```

void populateList()
{
    for (int i = 0; i < rows.Length / 3; i++)
    {
        ListViewItem templItem = new ListViewItem();
        templItem.Text = rows[i, 0];
        templItem.SubItems.Add(rows[i, 1]);
        templItem.SubItems.Add(rows[i, 2]);

        listSource.Items.Add(templItem);
    }
}

```

```

    itemsC = new ListViewItem[listSource.Items.Count];
    listSource.Items.CopyTo(itemsC, 0);
}

```

Thread Closed!