

Forum: [Hack Forums](#)

Thread: [.Net - Listbox Remove Duplicates / Remove Nulled Items?](#)

Gørilla - February 25, 2017, 6:39 pm

So I'm fairly well into my project and I'm very rarely getting errors now to where my listbox is giving null values (I believe its because of thread clashing if that's a thing), and I'm looking to try to find a way to remove duplicates.

I've looked up some tuts but none seem to ACTUALLY do anything.. There's code that looks like it'd do something, but nothing would happen.

At this point I have no raw code for the process but what I'm looking to do:

Listbox full of items -> click button called remove duplicates -> it removes duplicated items and the rest remain the same..

I don't care for a full spoon feeding, just a little help to further my knowledge. Thanks!

BahNahNah - February 25, 2017, 6:59 pm

You could use Distinct

[https://msdn.microsoft.com/en-us/library/bb348436\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb348436(v=vs.110).aspx)

Gørilla - February 26, 2017, 5:53 pm

(Feb 25, 2017 07:59 PM)BahNahNah Wrote: [\[CITE\]](#) You could use Distinct

[https://msdn.microsoft.com/en-us/library/bb348436\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb348436(v=vs.110).aspx)

I don't think that would be the way to go around this.

Closed User - February 26, 2017, 5:58 pm

(Feb 26, 2017 06:53 PM)Gørilla Wrote: [\[CITE\]](#) I don't think that would be the way to go around this. Prevent adding dupes in the first place or just don't use a listbox at all as you seem to use it for a functionality and not just GUI

PM me if you want any input

Gørilla - February 26, 2017, 6:00 pm

(Feb 26, 2017 06:58 PM)Big Benim Wrote: [\[CITE\]](#) Prevent adding dupes in the first place or just don't use a listbox at all as you seem to use it for a functionality and not just GUI

PM me if you want any input

Would you suggest using a rich text box or something?

Closed User - February 26, 2017, 6:33 pm

(Feb 26, 2017 07:00 PM)Gørilla Wrote: [\[CITE\]](#) Would you suggest using a rich text box or something?

I would suggest using just a list
ex.

```
Dim stringList as New List(Of String)
stringList.Add("thisisastring")
stringList.Add("thisisanothastring")
```

That would make everything way more smoother and as I read your past issues(Multithreading) it'll speed up the process a ton by doing this

```
Tasks.Parallel.ForEach(stringList, Function(line As String)
'Operation here
End Function)
```

SL x TnT - February 26, 2017, 10:08 pm

(Feb 26, 2017 06:53 PM)Gørilla Wrote: [\[CITE\]](#) I don't think that would be the way to go around this. Depending on how the items are added, it could be one of the better ways. If you aren't going to grab by index, using a HashSet to prevent duplicates in the first place works too.

Whatever you do, don't use Contains on a List when you're adding a large amount in a short time span.

INCEPTION_V - February 27, 2017, 9:20 am

(Feb 25, 2017 07:59 PM)BahNahNah Wrote: [\[CITE\]](#) You could use Distinct [https://msdn.microsoft.com/en-us/library/bb348436\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb348436(v=vs.110).aspx)

This is exactly the way to go.

SL x TnT - February 27, 2017, 12:49 pm

(Feb 27, 2017 12:02 PM)Orka Wrote: [\[CITE\]](#) How come nobody recommended hashset? Handles everything.

[https://msdn.microsoft.com/en-us/library/bb359438\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb359438(v=vs.110).aspx)

It was mentioned in multiple forms. Actually, only 1 person didn't suggest the use of a HashSet.

Gørilla - February 27, 2017, 12:51 pm

(Feb 27, 2017 01:49 PM)SL x TnT Wrote: [\[CITE\]](#) It was mentioned in multiple forms. Actually, only 1 person didn't suggest the use of a HashSet.

I will look into using this:)

Ninja - February 27, 2017, 4:31 pm

If they are strings you can use string.IsNullOrEmpty. Also, it is better to prevent duplicates from entering (ie. using a HashSet) than filtering them afterwards.

Gørilla - February 27, 2017, 4:37 pm

(Feb 27, 2017 05:31 PM)Ninja Wrote: [\[CITE\]](#) If they are strings you can use string.IsNullOrEmpty. Also, it is better to prevent duplicates from entering (ie. using a HashSet) than filtering them afterwards.

I will see if I can implement this into my project:).

Snify3000 - February 28, 2017, 2:34 am

I'm not a .NET developer so my answer might not be accurate but should give you some ideas.

First of all when you add a large list, use a stringlist first before you output it to your GUI.

This way you can edit the list like removing the duplicates in the first place.

In Delphi (that's what I program with) a stringlist has a property which lets you set a duplicate flag that avoids adding duplicates in the list. I found this answer online:

<http://stackoverflow.com/questions/47752/remove-duplicates-from-a-listt-in-c-sharp>

After the list is ready, then output the results, this way it is the fastest.

Good luck and enjoy developing :)

Gørilla - February 28, 2017, 6:24 am

(Feb 28, 2017 03:34 AM)Snify3000 Wrote: [\[CITE\]](#) I'm not a .NET developer so my answer might not be accurate but should give you some ideas.

First of all when you add a large list, use a stringlist first before you output it to your GUI.

This way you can edit the list like removing the duplicates in the first place.

In Delphi (that's what I program with) a stringlist has a property which lets you set a duplicate flag that avoids adding duplicates in the list. I found this answer online:

<http://stackoverflow.com/questions/47752/remove-duplicates-from-a-list-in-c-sharp>

After the list is ready, then output the results, this way it is the fastest.

Good luck and enjoy developing :)

I think this relates to vb.net as well. I'm going to look into the hash reference they were talking about when I fix up my program today:).