

# Forum:[Hack Forums](#)

Thread: [Converting vb.net to C# \(Small issue\)](#)

**MODERN** - October 16, 2019, 4:15 pm

Trying to convert a bit of code that was vb.net to C#

My issue is the following:

Problematic Code:

```
Code private void saveBTN_Click(object sender, EventArgs e)
{
    var save = new SaveFileDialog();

    if (save.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        System.IO.StreamWriter sWriter;
        sWriter = new System.IO.StreamWriter(save.FileName);

        for (var i = 0; i <= listBox3.Items.Count - 1; i++)

            sWriter.WriteLine(listBox3.Items.Item(i));

        sWriter.Close();
    }
}

// This is our create list button.
private void createBTN_Click(object sender, EventArgs e)
{
    if (listBox1.Items.Count == 0)
        MessageBox.Show("No usernames.");
    else if (listBox2.Items.Count == 0)
        MessageBox.Show("No passwords.");
    else
        makeList();
}

public void makeList()
{
    for (var i = 0; i <= listBox1.Items.Count - 1; i++)
    {
        for (var o = 0; o <= listBox2.Items.Count - 1; o++)

            listBox3.Items.Add(listBox1.Items.Item(i) + ":" + listBox2.Items.Item(o));

        label2.Text = listBox3.Items.Count;
    }
}
```

<https://paste.ubuntu.com/p/bwk4bCCSHM/>

Any assistance would be appreciated :)

**MavenGradle** - October 16, 2019, 4:37 pm

Retrieve a listbox item by using the index on the Items property of the ListBox control like so:

Code listBox.Items[index]

Do note that you are not compliant to any naming convention and I'd suggest that you give your controls proper names. In addition to that, writing to and reading from files can cause exceptions which I suggest you catch and properly handle.

There's also a way to write asynchronously to files using:

Code StreamWriter.WriteLineAsync

<https://docs.microsoft.com/en-us/dotnet/api/system.io.streamwriter.writelineasync?view=netframework-4.8>

**xHeaven** - October 17, 2019, 2:26 am

Kinda cheating, but when I want to convert a VB.NET project to C#, I just open the VB.NET project in dnSpy and watch the source code as C# source. This saves me a lot of headaches.

**Paskowsky** - October 17, 2019, 2:51 am

(Oct 17, 2019 02:26 AM)xHeaven Wrote: [\[CITE\]](#) Kinda cheating, but when I want to convert a VB.NET project to C#, I just open the VB.NET project in dnSpy and watch the source code as C# source. This saves me a lot of headaches.

That will work on most cases but still manual work(replacing all references to "Microsoft.VisualBasic.dll" with real .net framework alternatives) must be done in order to have working c# code.

Regards,  
Paskowsky

**0xDEAD10CC** - October 17, 2019, 11:19 am

(Oct 17, 2019 02:51 AM)Paskowsky Wrote: [\[CITE\]](#) That will work on most cases but still manual work(replacing all references to "Microsoft.VisualBasic.dll" with real .net framework alternatives) must be done in order to have working c# code.

Regards,  
Paskowsky  
You don't know what you're talking about at all.

Why would you not be able to use Microsoft.VisualBasic.dll in C#? It's like any other .NET shared library, it's just not a default library for C# projects. Should you though? Probably not, but you still *can*.

**Paskowsky** - October 17, 2019, 11:47 am

(Oct 17, 2019 11:19 AM)0xDEAD10CC Wrote: [\[CITE\]](#) You don't know what you're talking about at all.

Why would you not be able to use Microsoft.VisualBasic.dll in C#? It's like any other .NET shared library, it's just not a default library for C# projects. Should you though? Probably not, but you still *can*.

I know I can.

It's full of legacy and unsupported code and I just don't like using it. Usually when converting from VB.net to C# alot of implicit things VB does are not "supported" by c# syntax.

I think nobody should use it unless required by particular reason.

**0xDEAD10CC** - October 17, 2019, 12:22 pm

(Oct 17, 2019 11:47 AM)Paskowsky Wrote: [\[CITE\]](#) I know I can.

It's full of legacy and unsupported code and I just don't like using it. Usually when converting from VB.net to C# alot of implicit things VB does are not "supported" by c# syntax.

I think nobody should use it unless required by particular reason.

What you fail to understand is that the code is already compiled (IL), therefore there's no reason why C# wouldn't support that dll. The public interfaces are basic enough to be supported by either language. Also, it is not full of legacy and unsupported code; I believe you're thinking about Microsoft.VisualBasic.Compatibility.dll for VB6.

The reason not to use the dll you mentioned is that some of the standard .NET classes are more efficient.

**M O D E R N** - October 17, 2019, 1:23 pm

It's just a simple combolist maker I'm slapping together

So I was able to get some stuff fixed now.

<https://paste.ubuntu.com/p/s5ctxtJRGV/>

Added to github

<https://github.com/ModernAesthetics/ComboMaker>

**realsunjester** - October 18, 2019, 1:48 am

Code label2.Text = listBox3.Items.Count.ToString()

also, original error says assembly reference, which means youre just missing a dll or maybe youre not using the correct (update) version? but it seems you fixed it, idk, i dont use .NET stuff much anymore

**M O D E R N** - October 18, 2019, 8:36 am

(Oct 18, 2019 01:48 AM)realsunjester Wrote: [\[CITE\]](#) Code label2.Text = listBox3.Items.Count.ToString()

also, originalerror says assembly reference, which means youre just missing a dll or maybe youre not using the correct (update) version? but it seems you fixed it, idk, i dont use .NET stuff much anymore

Thank you for the help!

Looks like I'm just about there, I'm not getting any errors in VS but this is not right:

Current code:

<https://paste.ubuntu.com/p/Bq5NhpNBGS/>

```
Code public void makeList()
{
    for (var i = 0; i <= listBox1.Items.Count - 1; i++)
    {
```

```

        for (var o = 0; o <= listBox2.Items.Count - 1; o++)

            listBox3.Items.Add(listBox1.Items.IndexOf(i) + ":" + listBox2.Items.IndexOf(o));

        label2.Text = listBox3.Items.Count.ToString();
    }
}

```

**MavenGradle** - October 18, 2019, 12:05 pm

(Oct 18, 2019 08:36 AM)M O D E R N Wrote: [\[CITE\]](#) Thank you for the help!

Looks like I'm just about there, I'm not getting any errors in VS but this is not right:

Current code:

<https://paste.ubuntu.com/p/Bq5NhpnBGS/>

```

Code public void makeList()
{
    for (var i = 0; i <= listBox1.Items.Count - 1; i++)
    {
        for (var o = 0; o <= listBox2.Items.Count - 1; o++)

            listBox3.Items.Add(listBox1.Items.IndexOf(i) + ":" + listBox2.Items.IndexOf(o));

        label2.Text = listBox3.Items.Count.ToString();
    }
}

```

That's because you're not listening. I already told you how to retrieve a listbox item using an index yet you persist in using the wrong methodology. Let me explain. IndexOf returns -1 because it's looking for an item that is equal to the integer with the index value. Since the listbox does not contain that type and value, you're getting a "-1" to indicate that IndexOf could not find any results.

Again, you can select a listbox item by using the index property on the ObjectCollection like so:

Code listBox.Items[index]

Here's what the IndexOf method does exactly on an ObjectCollection:

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.listbox.objectcollection.indexof?view=netframework-4.8>

Read the first line and you'll know exactly why you're failing.

Quote:Returns the zero-based index where the item is located within the collection; otherwise, negative one (-1).

In addition to that, your makeList method should not be public. It should be private since it's not used anywhere else. For more information regarding access modifiers, please read (actually read) the following documentation:

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/access-modifiers>

Let me add something else. Your Git repository is not meant to store the obj/bin and bin/debug folders:

<https://github.com/ModernAesthetics/ComboMaker/tree/master/bin/Debug>

<https://github.com/ModernAesthetics/ComboMaker/tree/master/obj/Debug>

Thanks to that debug information, I now know that you store your C# projects in the following folder:  
Code D:\C#\ComboMakerAmong a few other things..

**M O D E R N** - *October 18, 2019, 12:47 pm*

Thanks for all the info, I think i have now fixed everything :)