

Forum:[Hack Forums](#)

Thread: [Form movement becoming slow](#)

_UnderDog - August 24, 2017, 7:24 pm

So I am still (slowly) developing an application which I released a Beta for a while ago (<https://hackforums.net/showthread.php?tid=5541855>) which requires the entire form to slide from right to left.

The problem with this is the form contains many controls, most of which are hidden until needed though when using the following code:

<https://pastebin.com/5G1pMqz3> (for some reason HF has a firewall against this code..)

It takes its time when loading, whereas if I compared it to an empty application, it is very smooth and instant.

I am still new to different concepts of coding but will converting everything to WPF help? This is the last thing I want to try as I'm using a theme and lots of controls. Any advice is appreciated!

DeadKube - August 24, 2017, 7:34 pm

I did not look into your code or anything you've posted containing any code but having many controls in a WinForms application is known to cause glitchy or laggy behaviour. WPF will solve this problem as it can handle many, many controls at once.

However, there's this property that winforms have called [doublebuffered](#) which might just solve the issue for you without having to convert to WPF. Although I will highly suggest that you DO convert to WPF since it's newer and allows for so many customization options in regards to GUI development. You'll thank your future self for learning it.

Edit: now I looked at your code. Please don't use `thread.sleep`. it will block your GUI!! [see this post for more information](#)..

_UnderDog - August 24, 2017, 8:35 pm

Thanks. Btw, it's multithreaded so it doesn't do that

KronosClouds - August 24, 2017, 10:04 pm

(Aug 24, 2017 07:24 PM)_UnderDog Wrote: [\[CITE\]](#) So I am still (slowly) developing an application which I released a Beta for a while ago SNIP which requires the entire form to slide from right to left.

The problem with this is the form contains many controls, most of which are hidden until needed though when using the following code:

SNIP

It takes its time when loading, whereas if I compared it to an empty application, it is very smooth and instant.

I am still new to different concepts of coding but will converting everything to WPF help? This is the last thing I want to try as I'm using a theme and lots of controls. Any advice is appreciated!

Try this instead maybe makes everything run asynchronous

```
Code Public Shared Async Sub Slideina(ByVal F As Form, s As Integer)
    Dim screenWidth As Integer = Screen.PrimaryScreen.Bounds.Width
    For i = 0 To s
        Await Task.Delay(3)
        F.Location = New Point(screenWidth - i, 0)
    Next
End Sub
```

Barret - August 24, 2017, 10:09 pm

Suspend the layout while you do the changing, resume when ur done.

[https://msdn.microsoft.com/en-us/library/system.windows.forms.control.suspendlayout\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.control.suspendlayout(v=vs.110).aspx)

_UnderDog - August 24, 2017, 10:26 pm

(Aug 24, 2017 10:04 PM)KronosClouds Wrote: [\[CITE\]](#) Try this instead maybe makes everything run asynchronous

```
Code Public Shared Async Sub Slideina(ByVal F As Form, s As Integer)
    Dim screenWidth As Integer = Screen.PrimaryScreen.Bounds.Width
    For i = 0 To s
        Await Task.Delay(3)
        F.Location = New Point(screenWidth - i, 0)
    Next
End Sub
```

Although this moves at the same extremely slow rate if I were to reduce the wait to 0.3 it runs much more smoothly than sleep. Thanks! :)

EDIT: Realised it's as good as 0 or no delay line at all. Might have to some variable delays based on different system performance.

(Aug 24, 2017 10:09 PM)Barret Wrote: [\[CITE\]](#) Suspend the layout while you do the changing, resume when ur done.

[https://msdn.microsoft.com/en-us/library/system.windows.forms.control.suspendlayout\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.control.suspendlayout(v=vs.110).aspx)

Unfortunately, this made no difference to the situation - But I will use this for another feature. Thanks!

DBots - August 24, 2017, 11:35 pm

Instead of:

```
ByVal F As Form
```

Wouldn't it make sense to use:

```
ByRef f As Form
```

0xDEAD10CC - August 25, 2017, 7:23 pm

(Aug 24, 2017 08:35 PM)_UnderDog Wrote: [\[CITE\]](#) Thanks. Btw, it's multithreaded so it doesn't do that

Explain? You're not setting a particular property to false are you and invoking this from a separate thread which needs to join the UI thread?

I think we need to see how you are threading this because I highly doubt you're doing it properly if you think multi-threading by itself fixes application hang problems magically.

Rhez - August 25, 2017, 11:53 pm

(Aug 25, 2017 07:23 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Explain? You're not setting a particular property to false are you and invoking this from a separate thread which needs to join the UI thread?

I think we need to see how you are threading this because I highly doubt you're doing it properly if you think multi-threading by itself fixes application hang problems magically.

Probably there's something wrong in his threading.

0xDEAD10CC - August 26, 2017, 12:37 pm

(Aug 25, 2017 11:53 PM)Rhez Wrote: [\[CITE\]](#) Probably there's something wrong in his threading.

This is what I was already suggesting. He had Thread.Sleep on a method which deals with UI code, he's saying it doesn't freeze the form because he's multi-threading, but what he doesn't understand is that you obviously can't do cross thread operations like this without setting that value which is only intended for debugging purposes, and this would end up making the threading completely pointless in that case and probably would degrade the performance of the application more than not multi-threading at all, with all the thread context switching and joins.

_UnderDog - August 26, 2017, 5:19 pm

(Aug 26, 2017 12:37 PM)0xDEAD10CC Wrote: [\[CITE\]](#) This is what I was already suggesting. He had Thread.Sleep on a method which deals with UI code, he's saying it doesn't freeze the form because he's multi-threading, but what he doesn't understand is that you obviously can't do cross thread operations like this without setting that value which is only intended for debugging purposes, and this would end up making the threading completely pointless in that case and probably would degrade the performance of the application more than not multi-threading at all, with all the thread context switching and joins.

I would agree. I'm completely self-taught with vb.net and I don't spend very much time coding but atm I just run that code in a different thread, like a background worker I suppose. And it does make more sense to use "Await Task.Delay(x)"

Right now I'm redesigning the entire application in WPF to try and solve the problem. Thanks everyone for the help and feedback :>

Xogg - August 26, 2017, 10:09 pm

In the code you made the thread sleep? Is that the main thread?

_UnderDog - August 26, 2017, 11:45 pm

(Aug 26, 2017 10:09 PM)Dexuh Wrote: [\[CITE\]](#) In the code you made the thread sleep? Is that the main thread?

I used this code to run it in a different thread.

```
Code Dim thread As System.Threading.Thread
      thread = New System.Threading.Thread(sub or function)
      thread.Start()
```

0xDEAD10CC - August 27, 2017, 9:28 am

(Aug 26, 2017 11:45 PM)_UnderDog Wrote: [\[CITE\]](#) I used this code to run it in a different thread.

```
Code Dim thread As System.Threading.Thread
      thread = New System.Threading.Thread(sub or function)
      thread.Start()
```

No, see you don't understand that this won't work to update the UI in any way unless you're on that thread. Multi-threading the UI like this is worse than not multi-threading at all...