

Forum:[Hack Forums](#)

Thread: [Create new forms in new thread gives error while using timer in new form](#)

scriptsandapps - February 14, 2022, 2:27 pm

Hello, i want to create new forms and be able to use it like when the visual studio adds the forms, but programmatically..

Now trying this. I get the idea normally the forms all run in the same thread??.

How can i make it all in 1 thread instead of my method but still add and delete them.. without freezing form1

[>>> Github ptoject Link <<<](#)

[>>>Download main.zip <<<](#)

<https://github.com/ScriptsAndApps/NotWorking/blob/main/Form1.cs>

Code List<Thread> jt = new List<Thread>();

```
.....
Thread newThread = new Thread(() =>
{
    Application.EnableVisualStyles();
    Application.Run(jf[tl]); /////// this keeps thread running and form
    jt[tl] = null; //clear thread after close
    jf[tl] = null; //clear item in list after close
});
```

<https://github.com/ScriptsAndApps/NotWorking/blob/main/SQUAREFORM.cs>

Code public SQUAREFORM()

```
{
    InitializeComponent();
    new Thread(() =>
    {
        while (!passthis.CloseApp)
        {
            if (!last && passthis.clickDisable)
            {
                last = true;

                /*out of this thread error*/

                uint initialStyle = GetWindowLong(this.Handle, -20);
                SetWindowLong(this.Handle, -20, initialStyle | 0x80000 | 0x20);
                return;
            }
            else if (last && !passthis.clickDisable)
            {
                last = false;
                System.Windows.Forms.MessageBox.Show("Test");
            }
        }
    });
}
```

```

    }
}
Thread.Sleep(100);

}).Start();
}
https://github.com/ScriptsAndApps/NotWorking/blob/main/ROUNDFORM.cs

```

```

Code    private void timer1_Tick(object sender, EventArgs e)
{
    if (!last && passthis.clickDisable)
    {
        last = true;
        /*out of this thread error*/
        this.Close()

        /*out of this thread error*/
        uint initialStyle = GetWindowLong(this.Handle, -20);
        SetWindowLong(this.Handle, -20, initialStyle | 0x80000 | 0x20);
    }
    else if (last && !passthis.clickDisable)
    {
        last = false;
    }
}

```

Thanks

MavenGradle - February 14, 2022, 2:44 pm

What makes you think that having your UI on a single thread is bad? There's this practice called using background workers or asynchronous code processes for intense methods and multi-threading/parallelization of your resource-heavy code so that your UI does not freeze even if your UI runs in a single thread.

My suggestion to you would be to keep your UI and your resource-heavy code separate and NOT to separate your UI in multiple threads depending on how many forms you have because you'd still be freezing up one form or the other if the code is that heavy.

Here's something to get you started:

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/>

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.threading.dispatcher?view=windowsdesktop-6.0>

In addition, I took a quick look at your code. You need to work on your naming conventions and best practices when it comes to programming in C#.

This, for example (I've highlighted the bad parts):

```

Quote:private void button2_Click(object sender, EventArgs e)
{

```

```

    ROUNDFORM form = new ROUNDFORM();
    addform(form);
}

```

Method names should be in PascalCase and be 'useful' (button2_click doesn't say much)

Give variables a proper meaningful name (and you could just call your 'addform' method with new

RoundForm() directly as a parameter)
Class names in PascalCase
Etc.

Cheat sheet:

<https://github.com/ktaranov/naming-convention/blob/master/C%23%20Coding%20Standards%20and%20Naming%20Conventions.md>

Full standard design guidelines:

<https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/>

scriptsandapps - February 14, 2022, 2:53 pm

(Feb 14, 2022 03:44 PM)MavenGradle Wrote: [\[CITE\]](#) What makes you think that having your UI on a single thread is bad? There's this practice called using background workers or asynchronous code processes for intense methods and multi-threading/parallelization of your resource-heavy code so that your UI does not freeze even if your UI runs in a single thread.

My suggestion to you would be to keep your UI and your resource-heavy code separate and NOT to separate your UI in multiple threads depending on how many forms you have because you'd still be freezing up one form or the other if the code is that heavy. Here's something to get you started:

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/>

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.threading.dispatcher?view=windowsdesktop-6.0>

Thank, Now I have an issue that i have not such focus to think about how to implement this to my particular Project.

Can you show an example how to create a new form from form1? and keep it open? so form1 does not break. Then I can implement it. I have no clue what to google or what to look at in doc's to start a thread that does not act like Thread.sleep() while i call it.

like

```
FormTask frm = FormTask();  
private static Toast FormTask(int slices)  
{  
    //open the new form2  
    return new FormTask();  
}
```

MavenGradle - February 14, 2022, 2:58 pm

(Feb 14, 2022 03:53 PM)scriptsandapps Wrote: [\[CITE\]](#) Can you show an example how to create a new form from form1? and keep it open? so form1 does not break. Then I can implement it. I have no clue what to google or what to look at in doc's to start a thread that does not act like Thread.sleep() while i call it.

Here you go:

<https://social.technet.microsoft.com/wiki/contents/articles/34371.close-windows-forms-application-when-last-window-closed.aspx>

Or you could change your application entry code (usually in Program.cs in the [Main](#) method) to

something along the lines of this, if you add a clear and decisive way to exit your application as well so your application doesn't keep running when all your forms close (better way to do it in the link above):

```
Code new FrmMain().Show();  
Application.Run();
```

You can create and display a new form like so:

```
Code new FrmSecond().Show();
```

More info on how to separate UI and resource-heavy logic (using async/parallel programming):

<https://docs.microsoft.com/en-us/dotnet/csharp/async>

<https://docs.microsoft.com/en-us/dotnet/standard/parallel-programming/>

[Video: <https://www.youtube.com/watch?v=2moh18sh5p4>]

scriptsandapps - February 14, 2022, 3:01 pm

(Feb 14, 2022 03:58 PM)MavenGradle Wrote: [\[CITE\]](#) Here you go:

<https://social.technet.microsoft.com/wiki/contents/articles/34371.close-windows-forms-application-when-last-window-closed.aspx>

Or you could change your application entry code (usually in Program.cs in the [Main](#) method) to something along the lines of this, if you add a clear and decisive way to exit your application as well so your application doesn't keep running:

```
Code new MainForm().Show();  
Application.Run();  
Will try this thank you.
```

I did not know this works

```
Code addform(new SQUAREFORM());
```

MavenGradle - February 14, 2022, 3:09 pm

(Feb 14, 2022 04:01 PM)scriptsandapps Wrote: [\[CITE\]](#) I did not know this works

```
Code addform(new SQUAREFORM());  
Be sure to look at the naming conventions!
```

I notice you're quite new to the language and you might benefit from a plugin like Resharper to help you follow best practices. It's not a free product, but you should definitely give it a try and they're offering a free trial for the first 30 days of usage (but trust me, give it a try):

<https://www.jetbrains.com/resharper/>

There are also tools like SonarQube, Roslyn, StyleCop and Coverity that can greatly improve your code quality and programming style.

scriptsandapps - February 14, 2022, 8:04 pm

(Feb 14, 2022 04:09 PM)MavenGradle Wrote: [\[CITE\]](#) Be sure to look at the naming conventions!

I notice you're quite new to the language and you might benefit from a plugin like Resharper to help you follow best practices. It's not a free product, but you should definitely give it a try and they're offering a free trial for the first 30 days of usage (but trust me, give it a try):

<https://www.jetbrains.com/resharper/>

There are also tools like SonarQube, Roslyn, StyleCop and Coverity that can greatly improve your code quality and programming style.

For now i just terminate all threads. Great video! this makes so much things easy