

Forum:[Hack Forums](#)

Thread: [Cancel async task](#)

Lookdadon - January 23, 2022, 9:55 am

Hi, i made a program that run asynchronously and i added CancellationTokenSource to my code but realised that the way it cancel the task is by throwing an exception. Even in a try/catch/finally the problem is that it still throw an exception. After research and alot of readings, i learnt that this is how it work, using the canceltokensource stop the task by throwing an exception but when you run the .exe stopping throw an error and ask you to continue and ignore the cancel event or quit program...

Is there another way to stop your code to run once a stop button is pressed? I dont want to thread sleep or wait i wanna know if there is a way to straight stop your code. There must be a way to do it.

Thanks in advance

itcorp - January 23, 2022, 5:35 pm

(Jan 23, 2022 10:55 AM)Lookdadon Wrote: [\[CITE\]](#) Hi, i made a program that run asynchronously and i added CancellationTokenSource to my code but realised that the way it cancel the task is by throwing an exception. Even in a try/catch/finally the problem is that it still throw an exception. After research and alot of readings, i learnt that this is how it work, using the canceltokensource stop the task by throwing an exception but when you run the .exe stopping throw an error and ask you to continue and ignore the cancel event or quit program...

Is there another way to stop your code to run once a stop button is pressed? I dont want to thread sleep or wait i wanna know if there is a way to straight stop your code. There must be a way to do it.

Thanks in advance

It would be really helpful to share your code,so we can see your implementation and your problem.In my opinion you have chosen the right way but you didn't implement it well because this requires proper handling and proper handling of cancellation exceptions over the tasks you want to stop without being complete.

Keep up the hard work and I wish you all the best.

GruntStyl - January 23, 2022, 8:50 pm

Source code is needed to see if you implemented it wrong. You don't need to share your entire source code if you do not wish but at least the portion that you're needing help with.

0x9F - January 24, 2022, 12:44 am

Spoonfeed alert: This is an absolute simple example on how to cancel asynchronous tasks. You will need to call `ThrowIfCancellationRequested` once the cancellation is requested and catch the `Exception OperationCanceledException`.

In this example the Task is terminating itself.

```
Code using System;  
using System.Threading;  
using System.Threading.Tasks;
```

```

namespace TaskCancellationExample
{
    class Program
    {
        static void Main(string[] args)
        {
            new Program().EntryPoint(args);
        }

        private CancellationTokenSource cancellationTokenSource = new CancellationTokenSource();
        void EntryPoint(string[] args)
        {
            Task task = Task.Run(() =>
            {
                try
                {
                    Work();
                }
                catch (OperationCanceledException)
                {
                    Console.WriteLine("Task Cancelled");
                }
                finally
                {
                    Console.WriteLine("Exection completed");
                }
            }, cancellationTokenSource.Token);

            task.Wait();

            if (cancellationTokenSource.IsCancellationRequested)
                Console.WriteLine("Task was cancelled through cancellation token");

            cancellationTokenSource.Dispose();

            Console.WriteLine("Press any key to exit");
            Console.Read();
        }

        void CancelTask()
        {
            if (!cancellationTokenSource.IsCancellationRequested)
                cancellationTokenSource.Cancel();
        }

        void Work()
        {
            for (int i = 0; i < 5; i++)
            {
                if (cancellationTokenSource.IsCancellationRequested)
                    cancellationTokenSource.Token.ThrowIfCancellationRequested();

                Console.WriteLine("Task {0} is running: {1}", Task.CurrentId, i);
            }
        }
    }
}

```

```

        if (i == 3)
            CancelTask();
    }
}
}
}

```

Joseph Ducreux - January 24, 2022, 3:03 am

hwnd or (much ghetto) just manually terminate

Lookdadon - January 25, 2022, 8:17 am

Thx all for you replies and sorry for taking time to answer. Well i found how to do it but in another way. Its gonna be hard to expose the code as i edited it. I have no doubt it was implemented correctly(im not bragging) and the exception that was thrown was what it was supposed to be thrown but tbh i didnt like this way to end running code.

Its not about that i have a problem sharing the code. Its a simple bulk gmail maker that automatically retrieve phone verification code. I first had 2 steps 1- make registration and receive phone verification code, manually entering the code, push send code button and Step2-continue registration but i implemented a way to get the phone verification code automatically and the whole operation can be made automatically so no need to stop BUT

For ppl who are facing the same problem, i found a way to stop the code:

Remember you need to run async

Lets say you have a simple code with a Start and Stop button:

1-You instanciate a "loop code" at 0 and you make getter and setter so it will be available anywhere:

```
Code private int myLoopCode = 0;
```

2-In your method btnStop_Click:

```
Code SetMyLoopCode(0);
```

```
while(GetMyLoopCode() == 0)
```

```
{
```

```
Thread.Sleep(2000);
```

```
}
```

3-In your method btnStart_Click:

```
Code SetMyLoopCode(1);
```

```
...//code to execute...
```

goodrabbit - March 3, 2022, 3:53 am

Better using Threads with ThreadAbort(thread);

Lookdadon - March 7, 2022, 12:38 pm

(Mar 3, 2022 04:53 AM)goodrabbit Wrote: [\[CITE\]](#) Better using Threads with ThreadAbort(thread); thank you for your answer! I will give a try!