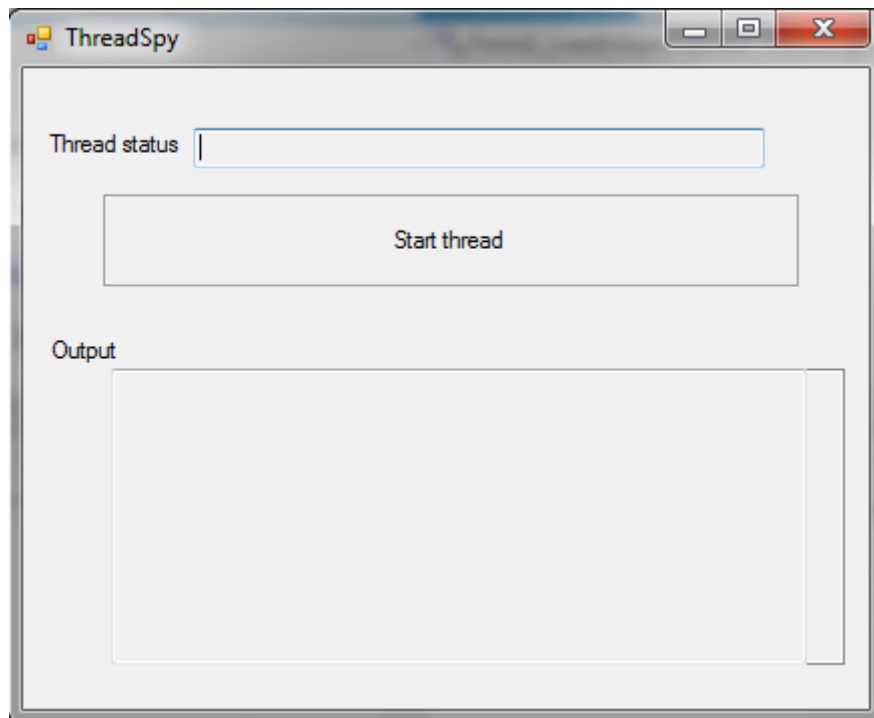


## RESOLUTION DU DEVOIR

Dans cette tâche, nous ferons des expériences avec plusieurs threads dans un processus.

Nous travaillerons avec le projet ThreadSpy.

Cette application à la fenêtre suivante:



Codes des références

Example du method Run1	Example du method Run2
<pre>public sub Run() for I As Integer = 0 to 19 Thread.Sleep(300) TextBoxHelper.AddChar(tb, c) Next End sub</pre>	<pre>public sub Run() for I As Integer = 0 to 19 Thread.Sleep(300) tb.Text += c; wrong, because this would generate an exception Next End sub</pre>

<pre>private sub ButtonStartThread_Click ()     textBox1.Text = "creation de processus en cours de 300 seconde";     drawingThread.Start () end sub</pre>	<pre>private sub timer1_Tick()     if (drawingThread &lt;&gt; null) this.TextBoxStatus.Text = drawingThread.ThreadState.ToString () end sub</pre>
---	---

DrawingRunnable dr = new DrawingRunnable(this.TextBoxOutput, c)
---

```
ThreadStart ts = new ThreadStart(dr.Run)
drawingThread = new Thread(ts)
```

### Question 1. Démarrez l'application.

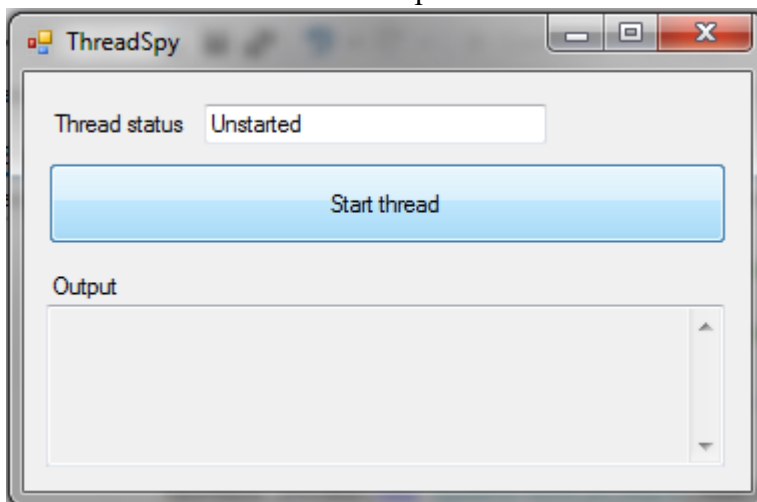
Lorsque vous cliquez sur le bouton Start thread, un nouveau fil est lancé qui dessine 20 caractères 'a' dans la zone de texte Sortie.

Vérifiez que l'état du thread C# est initialement «Non démarré».

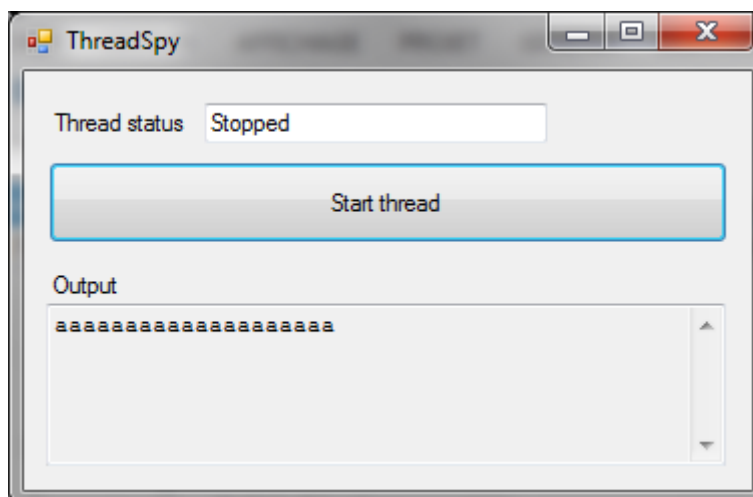
Cliquez maintenant sur le bouton «Start thread». Vous voyez que le nouveau thread est en cours d'exécution car il ajoute 20 caractères «a» à la zone de texte de sortie.

### Solution 1.

1. Interface d'un état en cours des processus



2. Interface d'un état de fin de traitement des processus



1. Code source modifié de l'exemple

A. Classe DrawingRunnable

```
Imports System.Threading
```

```
namespace ThreadSpy
```

```
class DrawingRunnable
```

```
    private md As TextBox  
    private a As char  
    private b As char  
    private c As char
```

```
    public sub DrawingRunnable(TextBox md,char a, char b, char c)
```

```
        this.kg = md  
        this.a = a  
        this.b = b  
        this.c = c  
    end sub
```

```
    public sub Run()
```

```
        for K As Int = 0 to 1  
            Thread.Sleep(300)  
            TextBoxHelper.AddChar(md, c,c,c)
```

```
        Next
```

```
        for K As Int = 0 to 2  
            Thread.Sleep(300)  
            TextBoxHelper.AddChar(md, b,b,b)
```

```
        Next
```

```
        For K As Int = 0 to 3  
            Thread.Sleep(300)  
            TextBoxHelper.AddChar(md, a, a, a)
```

```
    Next
```

```
    End sub
```

```
End sub
```

```
End class
```

B. classe TextBoxHelper

```
Imports System.Threading
```

```
Imports System.Windows.Forms;
```

```
namespace ThreadSpy
```

```
    class TextBoxHelper
```

```
        static private TextBox textbox
```

```
        public delegate sub UpdateTextCallback(char a, char b, char c)
```

```

static public sub AddChar(TextBox md, char a, char b, char c)
    textbox = tbx
    textbox.Invoke(new UpdateTextCallback(AddCharSave), a,b,c)

end sub
static private sub AddCharSave(char a, char b, char c)

    textbox.Text += b

end sub
end sub
end class

```

C. Code du method modifier

```

public sub Run()
    for K As Int = 0 to 3
        Thread.Sleep(300)
        TextBoxHelper.AddChar(md, c)
    Next

```

## Question 2. Utilisez le Gestionnaire des tâches pour afficher le nombre de threads du noyau dans cette application.

Démarrez le Gestionnaire des tâches et assurez-vous que la colonne 'Threads' est affichée (si nécessaire, utilisez Affichage → Sélectionner les colonnes...).

Fermez l'application et redémarrez-la.

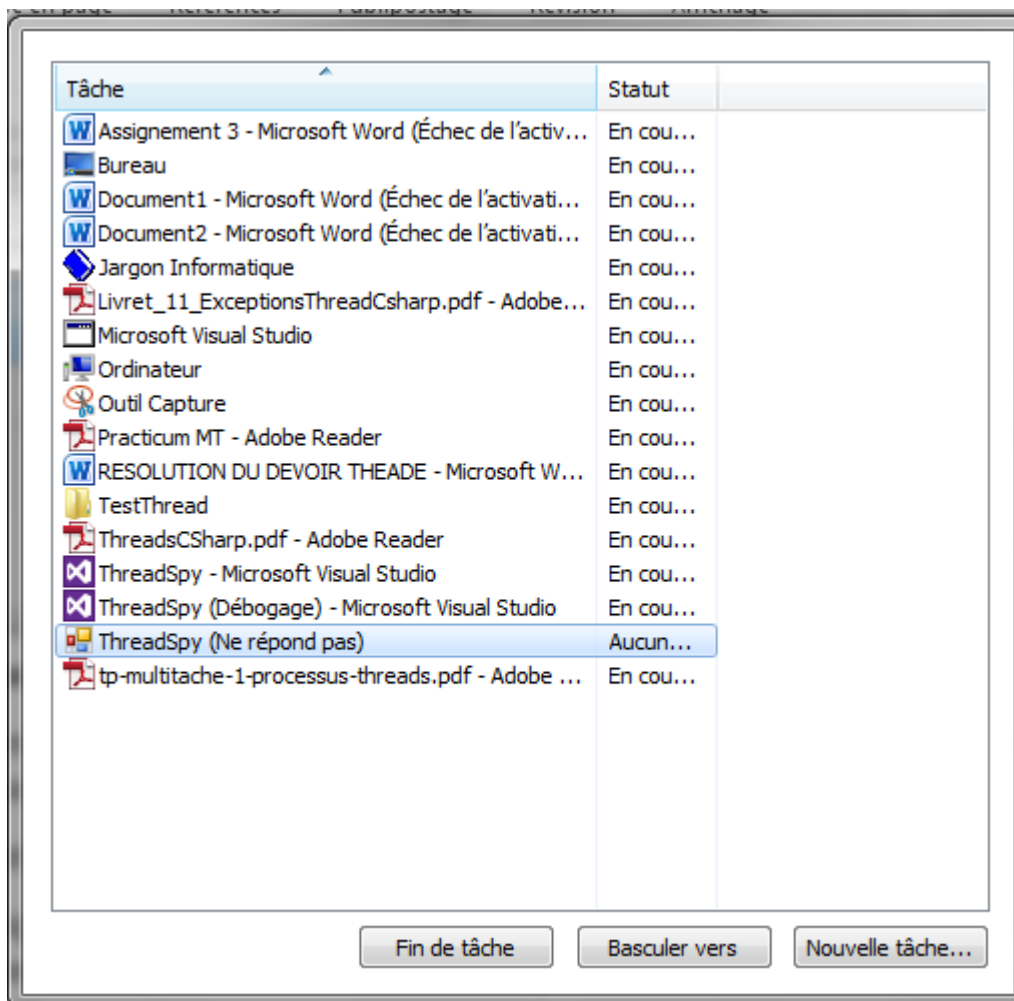
Recherchez le nombre de threads du noyau pour cette application dans le Gestionnaire des tâches.

Cliquez maintenant sur le bouton «Démarrer le fil».

Dans le Gestionnaire des tâches, vous pouvez voir qu'un thread de noyau supplémentaire est créé et que ce fil supplémentaire est supprimé lorsque le fil vb est «arrêté».

Cela signifie que le thread du noyau n'est créé que lorsque le thread vb est démarré et non lors de la création de l'objet Thread vb

## Solution 2.



### Question 3. Le thread AC # ne peut pas être démarré deux fois.

Pour vérifier cela, fermez l'application et redémarrez-la.

Cliquez sur le bouton "Démarrer le fil" et recommencez pendant que le fil est encore fonctionnement. Quel type d'exception est levé?

Fermez l'application et redémarrez-la.

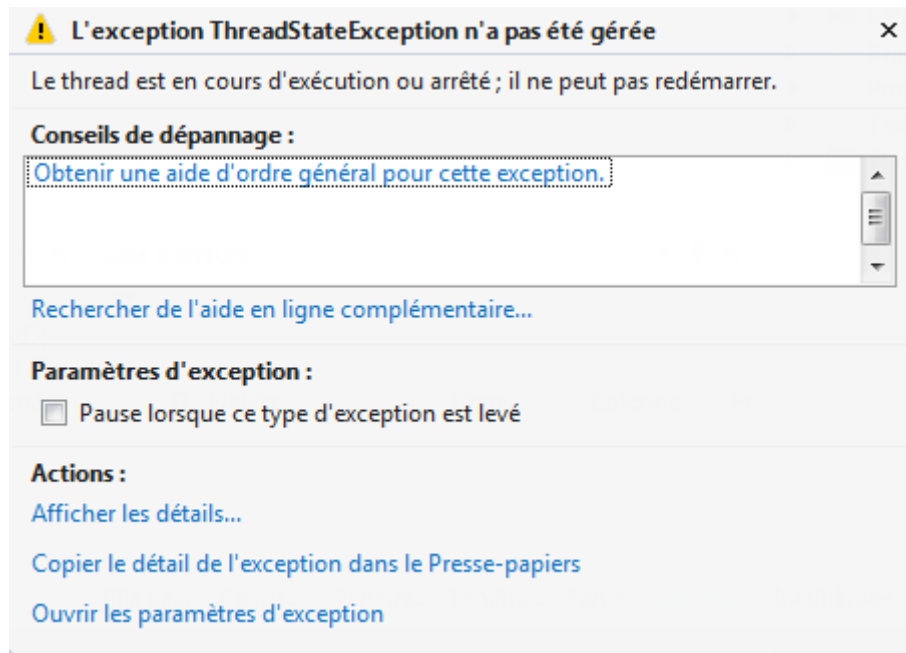
Cliquez sur le bouton «Démarrer le fil», et recommencez lorsque le fil est

"Arrêté". Quel type d'exception est levé?

### L'application ne s'arrête pas correctement.

Fermez l'application et redémarrez-la. Cliquez sur "Démarrer le fil" et arrêtez immédiatement l'application. Il ne se fermera pas correctement (peut-être que vous obtenez une exception).

Changez le nouveau fil de façon à ce qu'il se ferme immédiatement et correctement lorsque le l'application est fermée.



**Question 4. Changez le programme pour qu'un nouveau thread démarre à chaque fois lorsque vous cliquez sur le bouton «Démarrer le fil».**

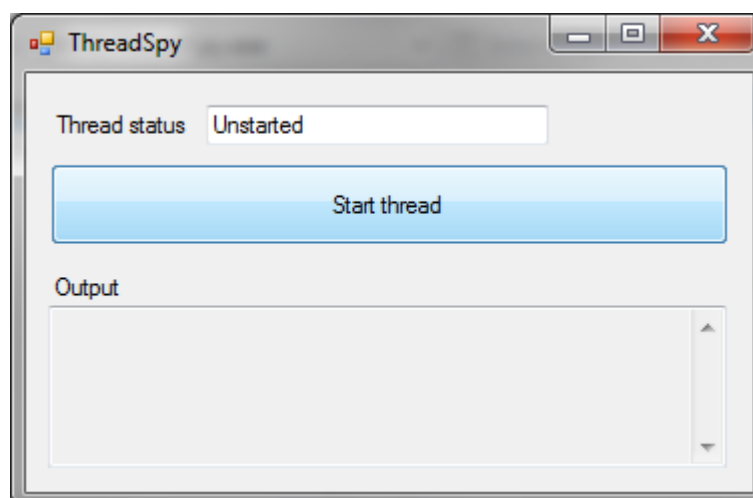
Pour voir la différence entre les fils, laissez chaque fil en imprimer un autre caractère dans la zone de texte: le premier fil doit imprimer 20 a, le deuxième fil

20 b, le troisième fil 20 c, et ainsi de suite.

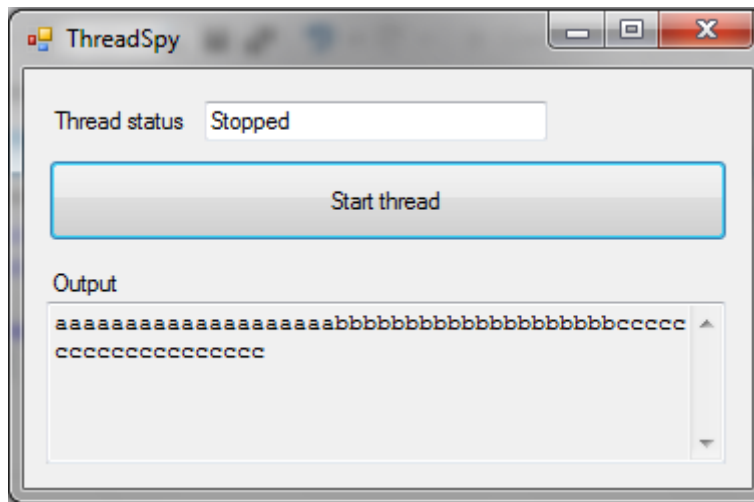
Bien entendu, le "Statut du thread" ne peut être affiché que pour un seul thread. Faites-le tel que toujours le statut du dernier thread créé est affiché. Vérifiez que vous pouvez cliquer maintenant sur " Start thread " plusieurs fois sans obtenir une exception.

**Solution 4.**

1. Interface d'un état en cours des processus



2. Interface d'un état de fin de traitement des processus



## 2. Code source modifié de l'exemple

```

public sub Run()
    for K As Int = 0 to 19
        Thread.Sleep(300)
        TextBoxHelper.AddChar(md, c,c,c)
    Next
    for K As Int = 0 to 19
        Thread.Sleep(300)
        TextBoxHelper.AddChar(md, b,b,b)
    Next
    for K As Int = 0 to 19
        Thread.Sleep(300)
        TextBoxHelper.AddChar(md, a,a,a)
    Next
End sub

```

### Question 5. Changez le programme de sorte que chaque nouveau fil en imprime un de plus caractère que le fil précédent.

- Le premier fil imprime 1 'a',
- le deuxième fil imprime 2 'b's,
- le troisième fil imprime 3 'c's,
- et ainsi de suite.

Pour ce faire, ajoutez un int NrOfChars privé à la classe

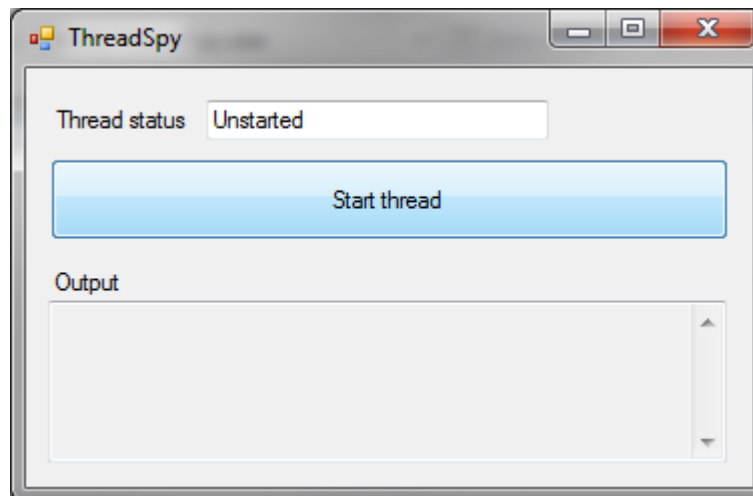
ThreadSpyForm et assurez-vous qu'il augmente à chaque fois qu'un nouveau le fil est créé. Testez votre application pour vous assurer qu'elle s'exécute toujours et se termine correctement.

Remarque: essayez d'appuyer plusieurs fois rapidement sur le bouton `` Démarrer le fil ", puis vous devriez toujours voir 1 'a', 2 'b's, 3' c's, etc. Ces caractères doivent être mélangés, mais le numéro de chaque caractère doit être comme ça.

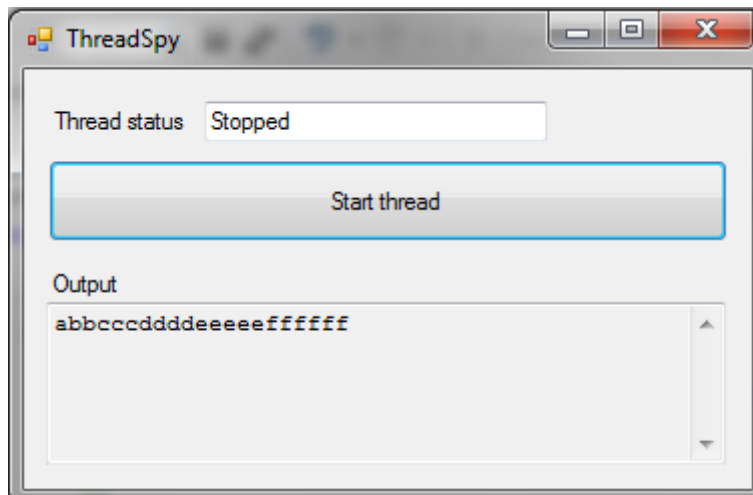
Par exemple: abcbccddeededeffffef.

## Solution 5.

1. Interface d'un état en cours des processus



2. Interface d'un état de fin des processus



3. Code source modifié de l'exemple

### A. Classe `TextBoxHelper`

```
Imports System.Threading
namespace ThreadSpy
```

```
class TextBoxHelper
```

```
static private TextBox textbox;
```

```
public delegate sub UpdateTextCallback(char a, char b, char c, char d, char e, char
```

```
f)
```

```
static public sub AddChar(TextBox md, char a, char b, char c, char d, char e, char f)
```

```
textbox = kg;
```

```
textbox.Invoke(new UpdateTextCallback(AddCharSave), a, b, c, d, e, f)
```

```
static private sub AddCharSave(char a, char b, char c, char d, char e, char f)
```



```

    {
        textbox.Text += b
    End sub

```

## B. Classe

```

Imports System.Threading
namespace ThreadSpy
    class DrawingRunnable
        private TextBox md
        private a As char
        private b As char
        private c As char
        private d As char
        private e As char
        private f As char

        public sub DrawingRunnable(TextBox md, char a, char b, char c, char d, char e,
char f)
            this.zb = md
            this.a = a
            this.b = b
            this.c = c
            this.d = d
            this.e = e
            this.f = f
        end sub

```

```

    public sub Run()

        for K As Int = 0 to 1
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md, f,f,f,f,f,f)
        Next
        for K As Int = 0 to 2
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md,e,e,e,e,e,e)
        Next
        for K As Int = 0 to 3
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md,d,d,d,d,d,d)
        Next
        for K As Int = 0 to 4
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md, c,c,c,c,c,c)
        Next
        for K As Int = 0 to 5
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md, b,b,b,b,b,b)
        Next

```

```
        for K As Int = 0 to 6
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md, a, a, a, a, a, a);
        Next
    End sub
```

### C. Méthode Run

```
public sub Run()
    for K As Int = 0 to 1
        Thread.Sleep(300)
        TextBoxHelper.AddChar(md, c,c,c)

        Next
        for K As Int = 0 to 2
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md, b,b,b)
        Next
        for K As Int = 0 to 3
            Thread.Sleep(300)
            TextBoxHelper.AddChar(md, a, a, a, a)
        Next

End sub
```