

7 ways to start a Task in .NET C#

 JANUARY 1, 2014  19 COMMENTS

New threads can be started using the Task Programming Library in .NET in – at last – 5 different ways.

You'll first need to add the following using statement:

```
1 | using System.Threading.Tasks;
```

The most direct way

```
1 | Task.Factory.StartNew(() => {Console.WriteLine("Hello Task library!"); });
```

Using Action

```
1 | Task task = new Task(new Action(PrintMessage));
2 | task.Start();
```

...where PrintMessage is a method:

```
1 | private void PrintMessage()
2 | {
3 |     Console.WriteLine("Hello Task library!");
4 | }
```

Using a delegate

```
1 | Task task = new Task(delegate { PrintMessage(); });
2 | task.Start();
```

Lambda and named method

```
1 | Task task = new Task( () => PrintMessage() );
2 | task.Start();
```

Lambda and anonymous method

```
1 | Task task = new Task( () => { PrintMessage(); } );
2 | task.Start();
```

Using Task.Run in .NET4.5

```
1 | public async Task DoWork()
2 | {
3 |     await Task.Run(() => PrintMessage());
4 | }
```

Using Task.FromResult in .NET4.5 to return a result from a Task

```
1 | public async Task DoWork()
2 | {
3 |     int res = await Task.FromResult<int>(GetSum(4, 5));
4 | }
5 |
6 | private int GetSum(int a, int b)
7 | {
8 |     return a + b;
9 | }
```

You cannot start a task that has already completed. If you need to run the same task you'll need to initialise it again.