

# System.Threading.ThreadState Enum

```
[ILAsm]
.class public sealed serializable ThreadState extends System.Enum

[C#]
public enum ThreadState
```

## Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Type Attributes:

- FlagsAttribute

## Summary

Specifies the execution states of a `System.Threading.Thread`.

## Inherits From: System.Enum

## Library: BCL

## Description

`System.Threading.ThreadState` defines the set of possible execution states for threads. Once a thread is created, it is in one or more of these states until it terminates. Not all combinations of `ThreadState` values are valid; for example, a thread cannot be in both the `System.Threading.ThreadState.Stopped` and `System.Threading.ThreadState.Unstarted` states.

The following table shows the actions that cause a thread to change state.

Action	ThreadState after Action
The thread is created	Unstarted
<code>System.Threading.Thread.Start</code> is invoked on the thread	Running
The thread calls <code>System.Threading.Thread.Sleep</code>	WaitSleepJoin

The thread calls <code>System.Threading.Monitor.Wait</code> to wait on an object	WaitSleepJoin
The thread calls <code>System.Threading.Thread.Join</code> to wait for another thread to terminate	WaitSleepJoin
The <code>System.Threading.ThreadStart</code> delegate methods finish executing	Stopped
Another thread requests the thread to <code>System.Threading.Thread.Abort</code>	AbortRequested
The thread accepts a <code>System.Threading.Thread.Abort</code> request	Aborted

In addition to the states noted above, there is also the `System.Threading.ThreadState.Background` state, which indicates whether the thread is running in the background or foreground.

The current state of a thread can be retrieved from the `System.Threading.Thread.ThreadState` property, whose value is a combination of the `System.Threading.ThreadState` values. Once a thread has reached the `System.Threading.ThreadState.Stopped` state, it cannot change to any other state.

## ThreadState.Aborted Field

```
[ILAsm]  
.field public static literal valuetype System.Threading.ThreadState  
Aborted = 0x100  
  
[C#]  
Aborted = 0x100
```

### Summary

The thread represented by an instance of `System.Threading.Thread` has terminated as a result of a call to `System.Threading.Thread.Abort`. A thread in this state is also in the `System.Threading.ThreadState.Stopped` state.

## ThreadState.AbortRequested Field

```
[ILAsm]  
.field public static literal valuetype System.Threading.ThreadState  
AbortRequested = 0x80  
  
[C#]  
AbortRequested = 0x80
```

### Summary

The `System.Threading.Thread.Abort` method has been invoked on the thread, but the thread has not yet received the pending `System.Threading.ThreadAbortException` that will attempt to terminate it.

# ThreadState.Background Field

```
[ILAsm]  
.field public static literal valuetype System.Threading.ThreadState  
Background = 0x4  
  
[C#]  
Background = 0x4
```

## Summary

The thread represented by an instance of `System.Threading.Thread` is being executed as a background thread, as opposed to a foreground thread. [*Note:* This state is controlled by setting the `System.Threading.Thread.IsBackground` property.]

# ThreadState.Running Field

```
[ILAsm]  
.field public static literal valuetype System.Threading.ThreadState  
Running = 0x0  
  
[C#]  
Running = 0x0
```

## Summary

The thread represented by an instance of `System.Threading.Thread` has been started and has not terminated.

To determine if a thread is running, check that its state does not include `System.Threading.ThreadState.Unstarted` and does not include `System.Threading.ThreadState.Stopped`.

## 1 ThreadState.Stopped Field

```
2 [ILAsm]  
3 .field public static literal valuetype System.Threading.ThreadState  
4 Stopped = 0x10  
  
5 [C#]  
6 Stopped = 0x10
```

## 7 Summary

8 The thread represented by an instance of `System.Threading.Thread` has terminated.

9

## 1 ThreadState.Unstarted Field

```
2 [ILAsm]  
3 .field public static literal valuetype System.Threading.ThreadState  
4 Unstarted = 0x8  
  
5 [C#]  
6 Unstarted = 0x8
```

## 7 Summary

8 The System.Threading.Thread.Start method has not been invoked on the thread.

9



## ThreadState.WaitSleepJoin Field

```
[ILAsm]  
.field public static literal valuetype System.Threading.ThreadState  
WaitSleepJoin = 0x20  
  
[C#]  
WaitSleepJoin = 0x20
```

### Summary

The thread represented by an instance of `System.Threading.Thread` is blocked as a result of a call to `System.Threading.Monitor.Wait`, `System.Threading.Thread.Sleep`, or `System.Threading.Thread.Join`.