

# Can a console app stay alive until it has finished its work?

Asked 16 years, 3 months ago   Modified 13 years, 3 months ago

Viewed 615 times



2



I've just read up on `Thread.IsBackground` and if I understand it correctly, when it's set to `false` the Thread is a foreground thread which means it should stay alive until it has finished working even though the app have been exited out. Now I tested this with a winform app and it works as expected but when used with a console app the process doesn't stay alive but exits right away. Does the `Thread.IsBackground` behave differently from a console app than a winform app?

`.net`

`multithreading`

Share

Improve this question

Follow

edited Aug 27, 2011 at 15:06



laurent

90.6k ● 82 ● 306 ● 438

asked Sep 20, 2008 at 4:55



John

5,461 ● 4 ● 32 ● 26

## 3 Answers

Sorted by:

Highest score (default)



2



The `Thread.IsBackground` property only marks if the thread should block the process from exiting. It doesn't perform any magic to keep the thread alive until some sort of explicit exit.

To quote the [Thread.IsBackground Property MSDN](#) (emphasis mine):

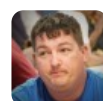
A thread is either a background thread or a foreground thread. Background threads are identical to foreground threads, except that background threads do not prevent a process from terminating. **Once all foreground threads belonging to a process have terminated, the common language runtime ends the process.** Any remaining background threads are stopped and do not complete.

In order to keep your console app alive you'll need to have some sort of loop which will spin until you ask it to stop via a flag or similar. Windows Forms applications have this built in because of their message pump (I believe).

Share Improve this answer

answered Sep 20, 2008 at 5:51

Follow



[Adrian Clark](#)

12.5k ● 5 ● 37 ● 42



1



I believe with the winforms based app you get a seperate thread to handle the messaging, so if the "main" thread exits, you still have a thread going to keep the process alive. With a console app, once main exits, unless you started a foreground thread, the process also terminates.



Share Improve this answer

answered Sep 20, 2008 at 4:58



Follow



[John Gardner](#)

25.1k ● 5 ● 61 ● 77

---

The process seems to terminate even though I've started a foreground thread. – [John](#) Sep 20, 2008 at 5:41

---



0



IMHO really you should be a lot more explicit about the expected semantics of your application and deliberately `<thread>.Join`.



Share Improve this answer

answered Sep 20, 2008 at 5:44

Follow



[stephbu](#)

5,082 ● 28 ● 42

