

Re: Thread.Abort()

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-10/msg03264.html>

- *From:* "Alan T" <alanpltseNOSPAM@xxxxxxxxxxxxxx>
 - *Date:* Thu, 19 Oct 2006 11:00:56 +1000
-

I partially understand your approach but not quite sure how to implement the stop and resume.

The syntax is not proper, I just show the meaning:

```
Class MyThreadWrapper
private _myThread
private enumerated type _state (pause, stop, start, resume)
method Pause
{ set _state = pause }
method Stop
{ set _state = stop }
method Start
{ set _state = start
_myThread.Start()
}
method Resume
{ set _state = resume }
```

If the user calls MyThreadWrapper.State(),
_state will be set to start and calls

constructor :
_myThread = new Thread(new ThreadStart(this.DoSomething));

In the _myThread, it executes DoSomething(), to do a loop, for example 50 times:

```
procedure DoSomething()
loop
Copy a file to a location
Update database record
Read the file content
Write the content to a log file
end loop
```

1) If the user calls MyThreadWrapper.Start()

Re: Thread.Abort()

2) If the user calls MyThreadWrapper.Pause(), how do I guarantee the loop stops at after the "Write the content to a log file" ? By checking the value of the _state?

```
procedure DoSomething()
loop
Copy a file to a location
Update database record
Read the file content
Write the content to a log file
while _state == pause
{
// no statement, so stay in this dummy while loop
}
end loop
```

2) What if the users calls MyThreadWrapper.Stop() ?

```
procedure DoSomething()
loop
Copy a file to a location
Update database record
Read the file content
Write the content to a log file
while _state == pause
{
// no statement
}
if _state == stop
{
exit;
}
end loop
```

Should it be like this:

```
procedure Stop()
{
_state = stop;
_myThread.abort();
}
```

Please comment

"Adityanand Pasumarthi" <AdityanandPasumarthi@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in message news:55E4782D-DE73-4EAB-841F-A481AC28039A@xxxxxxxxxxxxxxxxxxxx

Hi Alan,

Re: Thread.Abort()

Re: Thread.Abort()

Suspending and then Aborting a thread may cause data integrity problems if your thread in middle of updating some business data. And anyway these two methods are deprecated in .Net 2.0.

If your thread is doing some routine data related manipulations in a loop (or some loop kind of thing), then implement a small class that wraps your thread and provide "Pause", "Resume" and "Stop" methods on that class.

Define

your thread handler method as a method inside the class and based on the state of the class object ("Pause", "Resume" and "Stop") let the thread handler method pause, resume and stop by itself.

This way you gurantee the data safety in your application and also provide a decent way of pausing, resuming and stopping the work done by your thread.

--

Regards,
Aditya.P

"Alan T" wrote:

I will do several things in my thread:

Copy a file to a location

Update database record

Read the file content

Write the content to a log file

If I call Thread.Abort(), it may be possible to stop at the middle of the something ?