

## Re: Could Application.DoEvents(); messup code execution ??

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.general/2008-03/msg00345.html>

---

- *From:* "Kristijan Marin" <[kristijan.marin@xxxxxxxxxx](mailto:kristijan.marin@xxxxxxxxxx)>
  - *Date:* Thu, 20 Mar 2008 12:48:21 +0100
- 

Hi,

The flag would not exactly do the job, cause as i described .... I could see that entering method was done twice , didn't set trace on exit but Commit on DB was executed only ones ..I would expect Commit to be executed twice too.. and that is the second time I presume ... no Rollback was executed ... cause if it would then i would know and no RETURN was neither executed .....

Looks just like the code has lost it self somewhere in the middle .... and as I said in 1 or 2 examples form 2500.... so it's really hard to track ....

but if I can recall i thing that all this stuff started to happen when I included DoEvents() ..... so now I'll remove all that from code and hope for the best ...

Thanks,  
Kris

"Marc Gravell" <[marc.gravell@xxxxxxxxxx](mailto:marc.gravell@xxxxxxxxxx)> wrote in message <news:eAiUPaniIHA.1204@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Well, the Sleep isn't doing you any favors... you are on the same thread – you are just slowing it down (assuming there isn't any other threading going on).

DoEvents is notorious for causing re-entrancy issues exactly as you have described. If you are doing background work, I would recommend a worker thread, but this isn't convenient if you are talking to the UI lots (only the UI thread can talk to the UI controls; the worker thread would have to do lots of marshalling).

In this case, it sounds like the most pragmatic thing to do would be to add a flag (either a bool or an int counter) to prevent re-entrancy:

```
bool inProgress;  
void Some_Click(object sender, EventArgs args) {  
    if(inProgress) return; // dammit I heard you already! stop clicking!
```

Re: Could Application.DoEvents(); messup code execution ??

```
inProgress = true;  
try {  
  // your current code  
} finally {  
  inProgress = false;  
}  
}
```

Marc