

# Re: for loops

---

*Source:*

<http://www.tech-archive.net/Archive/Fox/microsoft.public.fox.programmer.exchange/2006-01/msg00341.html>

---

- *From:* "Nerd" <[Nerd@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Nerd@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Tue, 17 Jan 2006 11:27:04 -0800
- 

IF you are expecting the user to actually log in by typing in some information, then speed is irrelevant since it is strictly in the user's hands.

I heard the same lecture from a professor. Don't jump out of the code in the middle.

I personally would never use a FOR ENDFOR loop for something that I do not always expect to run a specified number of times. For a variable situation such as you are suggesting, I would always use a WHILE loop. As for there being more code lines or slower execution of a while loop vs a for loop, why would a good programmer care. Good programming is about clarity and reliability.

The following general structure has served me well in over fifty apps.

```
***** Start of main procedure
lsuccess = .F.
cntr = 1
DO WHILE !lsuccess and cntr <= 3
cntr = cntr + 1
lsuccess = <login_procedure>
ENDDO
IF lsuccess
< run app here >
ELSE
= MESSAGEBOX("Not allowed In",16,"Login Error")
ENDIF

***** End of Main procedure
```

"Carsten Bonde" wrote:

- > Andrew,
- >
- > a prof. at the university where i studied once said, using EXIT, "DO WHILE

Re: for loops

> ..T." and such constructs lets one think the programmer wasnt able to create  
> a proper termination of his loops. A statement i always kept in mind,  
> regardless of which programming-language i worked in.  
>  
> If speed is important, there actually may be situations where one may break  
> his own rules. Indeed a good point, thank you.  
>  
> Using EXIT (or LOOP) breaks the structured program-flow. Sometimes small  
> procedures grows up to real big ones, now facing an EXIT or LOOP in line 258  
> most likely confuses any person who didn't write the code (and even the  
> author 2 years later <gg>).  
>  
> Anyway – using welldefine termination of loops, or the easy "EXIT"-way might  
> be a matter of taste.  
>  
> --  
> Cheers  
> Carsten  
> \_\_\_\_\_  
>  
> "Andrew Howell" <ajh@work> schrieb im Newsbeitrag  
> [news:ecpE5x2GGHA.2300@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:ecpE5x2GGHA.2300@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
>> "Carsten Bonde" <bonde AT real-inkasso DOT de> wrote in message  
>> [news:uiTfLq0GGHA.344@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:uiTfLq0GGHA.344@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)  
>>> Beside that, i wouldnt use a FOR ENDFOR-construct, but a WHILE since you  
>>> should exit when the user enters a correct password. Using "EXIT" is in  
>>> my opinion NOT an option <s>  
>>  
>> I'm with you in principle but there \*are\* more lines of code for an  
>> equivalent WHILE and it does execute slower (I guess most likely because  
>> your increment happens in Fox rather than C.) It's unlikely the speed gain  
>> would be beneficial really (in any routine,) I'm just saying... <g>  
>>  
>> I've used FOR with EXIT for simple loops, eg checking a string for any  
>> alpha characters present (as soon as you hit one, may as well EXIT.)  
>> Although even that was before I knew how to do it with CHRTRAN.  
>> I've always hated DO WHILE .T. more than FOR..EXIT..NEXT – I guess because  
>> the construct is already waiting for a flag but for all that, why does  
>> EXIT exist? To not be used? <s>  
>>  
>> --  
>> Regards  
>> Andrew Howell  
>>  
>  
>  
>  
>  
>

- *Follow-Ups:*
  - ◆ *Re: for loops*
    - ◇ *From:* Andrew Howell
  
- *References:*
  - ◆ *RE: for loops*
    - ◇ *From:* Lee Mitchell
  - ◆ *Re: for loops*
    - ◇ *From:* Carsten Bonde
  - ◆ *Re: for loops*
    - ◇ *From:* Andrew Howell
  - ◆ *Re: for loops*
    - ◇ *From:* Carsten Bonde
  
- Prev by Date: *Re: detecting multiple monitors*
- Next by Date: *Re: Exe VERSION*
- Previous by thread: *Re: for loops*
- Next by thread: *Re: for loops*
- Index(es):
  - ◆ *Date*
  - ◆ *Thread*