

Re: Exit Do

Source:

<http://www.tech-archive.net/Archive/Scripting/microsoft.public.scripting.vbscript/2008-03/msg00500.html>

- *From:* "Anthony Jones" <Ant@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 24 Mar 2008 22:07:30 -0000
-

"Todd Vargo" <tlvargo@xxxxxxxxxxxxxxxxx> wrote in message
<news:uOyPeWXjIHA.6032@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Anthony Jones wrote:

Robbie Flower wrote:

I've been instructed by a dot net / asp programmer that Exit
Do is
not a great way to end a loop.

Can someone explain to me if this is so, or does this only
apply to
higher level programming languages ?

I am using nested loops and need to exit out of one loop into
another for something to apply. Any other functional ways of
doing
this ?

As with all things its best to consider sticking with good sound
principles rather than making rules. Exit Do/For/Function/Sub has
its uses but it also has its abuses.

Indiscriminate use of Exits can make it difficult to determine under
what conditions a particular block of code will end early. Also use
of Exits can make code more difficult to modify and/or lead to bugs
later on.

For example:–

```
Begin enumerating items
Do Until finished enumeration items
Code stage A
If Some Reason to stop loop then Exit Do
```

Re: Exit Do

Code Stage B Loop

In this code stage B is not be executed if there is a reason to exit the loop. What happens if a developer comes along 6 months later and adds some code to the loop prior to stage A that allocates a resource that needs to be explicitly released? It could be easy to miss the exit do in the existing code and simply add the releasing code after stage B. That would be a bug.

On the other you can find yourself jumping through all sorts of hoops trying to get execution of a loop to the end of the construct in order to avoid using an Exit. In which case this is also just as likely to lead to bugs and readability problems.

Therefore I would avoid Exits if the loop remains reasonably readable. If Exits are needed make them really, really obvious.

A good example where Exits would be useful, even expected, is in a search loop. The loop enumerates a set so it will end when the set is consumed. However it is expected that once a specific item is found it will end there and then. These loops tend to be simple and the presence of the exit is obvious.

Unfortunately, your reason/example is too vague to provide any worthwhile value. A developer who comes along 6 months later and introduces a bug by making modifications is the direct cause of the bug, not the original developers' EXIT usage which worked properly prior to modifications.

IMO, what you have described above is an unjustified excuse to blame a previous developer for incompetent modifications.

```
'Modification 6 months later (as described above)
Begin enumerating items
Do Until finished enumeration items
Modification requiring a resource release
Code stage A
If Some Reason to stop loop then
Modification to release resource
Exit Do
End If
Code Stage B
Loop
```

The code above is broken can you see why?

Its not about assigning blame to anyone. Its about realising that coding is only 20% communicating with a computer and 80% communicating with humans

Re: Exit Do

Re: Exit Do

(include a 6 month in future version of yourself). Humans are prone to error. The goal is to reduce the errors, not simply avoiding being the one blamed for the errors. Anything you can do when writing code that can avoid future potential bugs (whilst not increasing the risk of introducing bugs in the present) is a good thing.

Burying additional exit points are a source of bugs, whether you cause them or some other 'incompetent' programmer causes them.

According to your criteria I must be an 'incompetent' programmer. I'm sure at some point I've modified code that I thought I understood only for it to fail when I tested it. Tell me you've never modified an existing piece of code and it fail the first time you run it through? In many cases I've had to modify code that leaves a lot to be desired for clarity and its wasted a lot of my time when the effort to make the code clearer would have been minimal. Does the fact that the original code works of itself prove competency?

I think gave a fair and balanced view. IMO, additional exits may be a necessary evil and in some case they may even be the most elegant and expected solution. However if they can be reasonably avoided they should.

—

Anthony Jones – MVP ASP/ASP.NET

.