

Forum:[Hack Forums](#)

Thread: [For loop\(s\), Delayed Expansion](#)

bunny-Giles - January 29, 2016, 3:48 am

Hi,

I am a beginner with batch scripting and would be grateful for some advice...

Is it possible to use a for loop within a for loop in a batch script? - I need to loop through 1000 files and for each file repeat a 3 second cycle up to 300 times

To that end, how do I break out of the second for loop early?

Also, can you explain how delayed expansion works - i've seen it in scripts, but don't get what it does...would I need it here?

Any help will be gratefully received.

Thanks

jahwi - January 29, 2016, 5:08 am

You can use a for loop within a for loop. This is called a nested loop. You can type this:
Code FOR /?In an open command prompt window to get the full list of switches and uses.
Also, you should get a standard batch programming PDF to get you started.
As for exiting the loop early, That ranges between implausible and impossible.

When you're online, go here: [Tutorial](#) to learn batch scripting from this forum.

Welcome to batch.

?CARUs L?VE?™ - January 29, 2016, 5:21 am

```
Code for . . . ( . . . ) do (
    for . . . ( . . . ) do (
        if . . . equ . . . (
            do something
        )
    )
)
```

b00st3d - January 29, 2016, 4:57 pm

This is something I put together a while ago to demonstrate what delayed expansion does.

```
Code @echo off
setlocal enabledelayedexpansion
set x=1
echo +=====+
echo ^| No delayedexpansion ^|^| delayedexpansion ^|
echo ^|======^|
```

REM Batch parses a for loop as soon as it reaches the loop by default
REM this results in the variables that are in the loop not being updated x in this case
REM Although the end result is the same if you need to interact with x during the loop
REM then you will need to enable delayed expansion.

REM delayedexpansion causes batch to parse the value of the variable every time it comes across
REM the variable. In a loop this means the variable can be change it's value through the loop.
REM run this script to see the example below.

```
for /l %%a in (1,1,8) DO echo ^|   x = %x%           ^|^|   x = !x!   ^|& set /a x=x+1
echo ^|=====^|
echo ^|   x = %x%           ^|^|   x = !x!   ^|
echo +=====+
pause >nul
```

annikk.exe - January 30, 2016, 3:52 am

(Jan 29, 2016 04:48 AM)bunny-Giles Wrote: [\[CITE\]](#) Hi,

I am a beginner with batch scripting and would be grateful for some advice...

Welcome! ::purr::

You have found a good place here. Lots of friendly people/cats to play and learn with. :>

Quote:Is it possible to use a for loop within a for loop in a batch script?

Yes it is! Although, there are a couple of different methods...

Quote: - I need to loop through 1000 files and for each file repeat a 3 second cycle up to 300 times

Are all of the files in the same folder?

If that's the case, you can get the list of 1000 files using dir /b /a:-d

(Dirbad is actually the name of a pirate, true story..)

The /b bit means "use basic formatting"

The /a:-d part is for selecting "attributes" of the files we want, in this case, we want only files that are NOT directories (hence "-d" instead of "d")

Try typing dir /b /a:-d at a command prompt so you can see what this command does. :>

Then you need to loop over the files...

So I would suggest to use a for /f loop to iterate over each entry in the dir /b /a:-d output.

Since we don't want the loop to split the line up into tokens, we should also specify that there are no delimiters.

```
Code for /f "delims=" %%a in ('dir /b /a:-d') do (
    echo Oki now I am processing %%a
)
```

Are you following along so far? :> This is the outer part of the loop. We can cover the inner part too, but let you know if you have any questions about this so far.. :>

Quote: To that end, how do I break out of the second for loop early?

Spoiler alert... we use a GOTO statement. :P

Quote: Also, can you explain how delayed expansion works - i've seen it in scripts, but don't get what it does... would I need it here?

You might need it here, depending on how you choose to do your loops.

Okay... here goes my Delayed Expansion explanation.

In batch, normally we refer to variables like %variable%

If we enable Delayed Expansion, we can also refer to that same thing as !variable!

The exclamation marks are a different set of symbols, which also denote a variable.

Example:

Code @echo off

setlocal EnableDelayedExpansion

set "var=Cats are best"

echo %var%

echo !var!

pause

Code Cats are best

Cats are best

Press any key to continue . . .

Okay, so hopefully this gets you comfortable with the idea that !variable! == %variable%

Now, the question is, what does this let you do?

Well, the difference comes with the way the variables are evaluated.

The crux is that percentages are evaluated when the interpreter reaches that line whereas exclamations are evaluated when the interpreter reaches the variable

So for example, when we run a line of code like this:

Code echo %var%

echo !var!

First batch evaluates any percentage-variables to their actual values, and the line becomes this:

Code echo Cats are best

echo !var!

So that is why it's called "Delayed Expansion"

Because exclamation variables are not evaluated when the line is reached.. their evaluation is "delayed" until the last possible moment. :>

Delayed Expansion is very useful for creating array-like behaviour, or for iterating over characters in a string, and for a number of other advanced things.

Quote:Any help will be gratefully received.
Thanks

And likewise, we will fall all over ourselves to help you out, when you show the slightest interest in learning.. :> ::purr::

Zystem Fryar - January 30, 2016, 10:37 pm

Learn to use :Labels,
learn to Call them,
learn to end sections with goto :EOF
and learn to break out of them when you need to with the same...

bunny-Giles - January 31, 2016, 9:10 am

Thank you for all your responses:

Particular thanks to annikk.exe for the detailed break down; b00st3d for the helpful demonstration and jahwi for pointing me to the tutorial.

?CARUs L?VE?™ - January 31, 2016, 10:00 am

<https://hackforums.net/showthread.php?tid=3645708>

Zystem Fryar - January 31, 2016, 10:06 am

(Jan 31, 2016 11:00 AM)?CARUs L?VE?™ Wrote: [\[CITE\]](#)

<https://hackforums.net/showthread.php?tid=3645708>

Good one :)

Nearly got all my tricks in there :)

?CARUs L?VE?™ - January 31, 2016, 11:26 am

(Jan 31, 2016 11:06 AM)Zystem Fryar Wrote: [\[CITE\]](#) Good one :)

Nearly got all my tricks in there :)

Could you clarify on "your" tricks?

Zystem Fryar - January 31, 2016, 8:57 pm

(Jan 31, 2016 12:26 PM)?CARUs L?VE?™ Wrote: [\[CITE\]](#) Could you clarify on "your" tricks?

Well, within batch, Some specifics with how I handle Tokens, and their expansion. and assigning them to variables before a call statement (to avoid shifts in the Label)

example:

```
Code FOR /F "tokens=1-14 delims=;" %%a IN ("CtrlLine_.lst") DO set vr1_a=%%a && set  
vr1_b=%%b && set vr1_c=%%c && set vr1_d=%%d && set vr1_e=%%e && set vr1_f=%%f  
&& set vr1_g=%%g && set vr1_h=%%h && set vr1_i=%%i && set vr1_j=%%j && set vr1_k=%%k  
&& set vr1_l=%%l && set vr1_m=%%m && set vr1_n=%%n && call :Label ....  
or in some cases
```

```
Code FOR /F "tokens=1-14 delims=;" %%a IN ("CtrlLine_.lst") DO (  
set vr1_a=%%a  
set vr1_b=%%b  
set vr1_c=%%c  
set vr1_d=%%d  
set vr1_e=%%e
```

```

set vr1_f=%%f
set vr1_g=%%g
set vr1_h=%%h
set vr1_i=%%i
set vr1_j=%%j
set vr1_k=%%k
set vr1_l=%%l
set vr1_m=%%m
set vr1_n=%%n
call :Label ....
)

```

?CARUs L?VE?™ - February 1, 2016, 8:27 am

(Jan 31, 2016 09:57 PM)Zystem Fryar Wrote: [\[CITE\]](#) Well, within batch, Some specifics with how I handle Tokens, and their expansion. and assigning them to variables before a call statement (to avoid shifts in the Label)

example:

```

Code FOR /F "tokens=1-14 delims=;" %%a IN ("CtrlLine_.lst") DO set vr1_a=%%a && set
vr1_b=%%b && set vr1_c=%%c && set vr1_d=%%d && set vr1_e=%%e && set vr1_f=%%f
&& set vr1_g=%%g && set vr1_h=%%h && set vr1_i=%%i && set vr1_j=%%j && set vr1_k=%%k
&& set vr1_l=%%l && set vr1_m=%%m && set vr1_n=%%n && call :Label ....
or in some cases

```

```

Code FOR /F "tokens=1-14 delims=;" %%a IN ("CtrlLine_.lst") DO (
set vr1_a=%%a
set vr1_b=%%b
set vr1_c=%%c
set vr1_d=%%d
set vr1_e=%%e
set vr1_f=%%f
set vr1_g=%%g
set vr1_h=%%h
set vr1_i=%%i
set vr1_j=%%j
set vr1_k=%%k
set vr1_l=%%l
set vr1_m=%%m
set vr1_n=%%n
call :Label ....
)

```

Along with the improper use of '&&', it still doesn't explain the difference between how these are YOUR tricks, and just the syntax of the language.

jahwi - February 1, 2016, 3:08 pm

(Feb 1, 2016 09:27 AM)?CARUs L?VE?™ Wrote: [\[CITE\]](#) Along with the improper use of '&&', it still doesn't explain the difference between how these are YOUR tricks, and just the syntax of the language.

...more stuff about Icarus going after Zystem...

Remember that the OP is a FOR-loop beginner. Stop posting code snippets that the OP does not understand. @Icarus good one tho.

