

Forum:[Hack Forums](#)

Thread: [Checking command-line variable vs a variable on an online document](#)

DoxinYourBotnetM8 - March 16, 2015, 1:37 pm

If I have a script

```
Code @echo off
var1=var1
var2=var2
var3=incorrectvar
```

Is there any way I could check those variables against a text document on a website?

The text document (or any file extension, it doesn't really matter) would list the variables like;

```
Code var1=var1
var2=var2
var3=var3
```

Therefore var1 and var2 would print success messages whilst var3 would print an error message.

Essentially, I'm trying to make it so I can give people premium keys to my program, but the program will only allow this premium key if it's listed on the site.

Example script:

```
Code @echo off
set key=goodkey1
```

Example site file:

```
Code goodkey1
goodkey2
```

and since goodkey1 matches with something on the site, the premium access would be allowed.
But if the script was as follows;

```
Code @echo off
set key=badkey
```

Since "badkey" isn't listed on the site file the program would deny premium access.
Is this possible?

Nakamoto. - March 16, 2015, 5:39 pm

In order to check it you would have to download the file to local storage on your PC (I am pretty sure this can be done with a command like bitsadmin, if not vbs) then check for the variable.

DoxinYourBotnetM8 - March 16, 2015, 6:38 pm

(Mar 16, 2015 05:39 PM)RenRioku™ Wrote: [\[CITE\]](#) In order to check it you would have to download the file to local storage on your PC (I am pretty sure this can be done with a command like bitsadmin, if not vbs) then check for the variable.

I know there is a way to download files (I've used wget before) but I was hoping I could do it all online so I could make it as secure as possible.

However it seems as if that'll have to work.

White Light - March 16, 2015, 7:27 pm

(Mar 16, 2015 06:38 PM)DoxinYourBotnetM8 Wrote: [\[CITE\]](#) I know there is a way to download files (I've used wget before) but I was hoping I could do it all online so I could make it as secure as possible.

However it seems as if that'll have to work.

Not with just batch, sadly. You could possibly POST the variables with WGET and then analyze the response, but that's beyond my knowledge.

b00st3d - March 16, 2015, 7:47 pm

If you want to be secure, don't use batch. Anyone can look at your batch file and get the address for the document you are comparing to. For me what comes to mind would be java to do this just because I have been working on that a little bit. Java will allow you to pull data from a text file that resides on the internet and allow you to manipulate that data as needed. I am not strong enough with java to give anymore specifics than that but it is a starting point.

annikk.exe - March 17, 2015, 3:40 am

Oh come on, this IS possible in pure batch and you guys should know it by now!

BITS will allow you to download the webpage.

for /f lets you analyse the text file you downloaded.

If statements to check for equality.

Boom!

b00st3d - March 17, 2015, 5:30 am

(Mar 17, 2015 03:40 AM)annikk.exe Wrote: [\[CITE\]](#) Oh come on, this IS possible in pure batch and you guys should know it by now!

BITS will allow you to download the webpage.

for /f lets you analyse the text file you downloaded.

If statements to check for equality.

Boom!

Yes possible but not secure as he is asking. A programming language where the script is compiled and not visible would give an extra layer of security. If the user can read the text file that is being accessed by the script then they will know the key or whatever it is to unlock his program.

annikk.exe - March 17, 2015, 11:10 am

Yeah, it's woefully insecure, that is true. Totally possible though. :>

DoxinYourBotnetM8 - March 17, 2015, 1:01 pm

Technically couldn't I just have the document permissions hidden in a .HTACCESS file?

That way it could still be downloaded, but not viewed.

And afterwards I could just have the program delete the file so they wouldn't be able to view it on their desktop.

I could also do some sort of really shitty encryption system;

Code set a=www

```
set b=coldgaming.net
set c=file
set d=key
set e=.
set f=/
set g=http
set h=:
```

%g%%h%%f%%f%%a%%e%%b%%f%%d%%c%
(or something similar, I might've messed up)

annikk.exe - March 17, 2015, 1:56 pm

Nah, it's because somewhere in your batch file you're going to have something like:

Code if %keyValid% == true call :launchPremiumProgram
The user can simply edit it to:

Code if 1 == 1 call :launchPremiumProgram
Obfuscation is one option, but it's tricky to do it well unless you're an expert.

DoxinYourBotnetM8 - March 17, 2015, 2:16 pm

(Mar 17, 2015 01:56 PM)annikk.exe Wrote: [\[CITE\]](#) Nah, it's because somewhere in your batch file you're going to have something like:

Code if %keyValid% == true call :launchPremiumProgram
The user can simply edit it to:

Code if 1 == 1 call :launchPremiumProgram
Obfuscation is one option, but it's tricky to do it well unless you're an expert.
Ay, I forgot the user can edit the program himself..

How would I go about converting my program from Batch to java?

WOLGA - March 17, 2015, 3:21 pm

(Mar 17, 2015 02:16 PM)DoxinYourBotnetM8 Wrote: [\[CITE\]](#) Ay, I forgot the user can edit the program himself..

How would I go about converting my program from Batch to java?
Hm... I'll tell you once you tell me how to turn my Java into ASM. (It's a rhetorical question)

b00st3d - March 17, 2015, 6:36 pm

(Mar 17, 2015 02:16 PM)DoxinYourBotnetM8 Wrote: [\[CITE\]](#) Ay, I forgot the user can edit the program himself..

How would I go about converting my program from Batch to java?
Here is a good place to start.
[programmer's Heaven](#)

afaik there is no software to convert batch to java but it is a good language to learn as it is cross platform and widely used.

Llamafaggot - March 18, 2015, 3:37 am

(Mar 17, 2015 02:16 PM)DoxinYourBotnetM8 Wrote: [\[CITE\]](#) Ay, I forgot the user can edit the

program himself..

How would I go about converting my program from Batch to java?

Not possible due to how the two languages operate.

Batch is a series of commands ran one-after-another, each command is either a subroutine of the actual interpreter (the more simple commands AFAIK), or an external program the interpreter knows the location of (more complex commands AFAIK). Batch is an interpreted language, that is the interpreter reads over your script and executes it line-by-line, each line is read by the interpreter and the interpreter does some other code to fulfill the purpose of your command.

Java is an object-oriented language, data and elements you work with are represented as objects in memory, the class you write your code in is represented as an object stored on a stack. Each "command", "instruction" or "keyword" in Java edits the stack or an item on the stack (not in that way, this way implies that it's linear, the object is pushed to the stack then edited, it's more so the object is built based on the data you give in the form of "instructions" or "keywords" and then is pushed to the stack). Java is a compiled language, when you write your source code, to execute it you must compile it to JVM bytecode, which is so similar to the code your CPU executes natively (machine code, assembly, ASM), but a little different (JVM bytecode contains instructions that the CPU does not understand, so the JVM interprets the bytecode and translates it to a native language the CPU can understand, this is how Java is platform-independent (works on any system regardless of OS as long as you have a working JRE installed)).

The two languages operate worlds different from each other, have differing models for how they process instructions, handle variables different (in Batch a variable is basically a "keyword" which gets replaced by a value, in Java a variable on the stack is a reserved index on the stack for a value to be pushed to, same goes for in the memory), and overall are worlds different.

So you cannot translate from Batch to Java, at all. Not possible. You'll need to learn Java and write this from scratch I'm afraid.

DoxinYourBotnetM8 - March 18, 2015, 1:31 pm

I wasn't talking about converting the file over just like that. I know I'll need to learn Java. I was just hoping you could give me some references for similar Java code that I could work with (eg: what would the equivalent of echo be?)

I would hope it wouldn't be as simple as a program that can convert batch to Java. That'd take the fun out of it ;)

White Light - March 18, 2015, 5:41 pm

(Mar 18, 2015 01:31 PM)DoxinYourBotnetM8 Wrote: [CITE](#) I wasn't talking about converting the file over just like that. I know I'll need to learn Java. I was just hoping you could give me some references for similar Java code that I could work with (eg: what would the equivalent of echo be?)

I would hope it wouldn't be as simple as a program that can convert batch to Java. That'd take the fun out of it ;)

```
Echo -> System.out.println("OUTPUT MESSAGE");
```

Llamafaggot - March 18, 2015, 11:18 pm

(Mar 18, 2015 01:31 PM)DoxinYourBotnetM8 Wrote: [CITE](#) I wasn't talking about converting the file over just like that. I know I'll need to learn Java. I was just hoping you could give me some references for similar Java code that I could work with (eg: what would the equivalent of echo be?)

I would hope it wouldn't be as simple as a program that can convert batch to Java. That'd take the fun out of it ;)

Exactly what White said, but it's not so much converting Batch commands to Java code, you should learn how the Java code actually works, rather than what the equivalent code is. This is because both languages are fundamentally different.

The `println()` method belongs to the `out PrintStream` object type, which belongs to the `System` class statically. You should know what these are and what the language I just used means.

For reference, much of the same programming concepts are the same, IF statements are relatively the same, but the syntax and possibilities are completely different. In Batch, an IF statement would be:

```
Code set someVar=Hello World
```

```
if "%someVar%"=="Hello World" (  
    echo True
```

```
)Whereas in Java, it would be this within a method:
```

```
Code String someVar = "Hello World";
```

```
if (someVar.equals("Hello World")) {  
    System.out.println("True");  
}
```

```
// Or, more inefficiently...
```

```
if (someVar == "Hello World") {  
    System.out.println("True");  
}
```

In Batch, you can only have one binary condition, true or false, and it is only either a comparison between two strings (technically speaking all variables in Batch are strings) or two integers. The condition must be specified for the IF statement to be valid, you must specify that `val1 == val2`. In Java, on the other hand, it handles this functionally. You don't need a condition directly provided, all you need is a binary value to use, whether it be provided by a condition or method call is up to you.

Java also uses binary logic, the use of AND and OR operators can be used to "collapse" two separate binary values down to one, for example, ANDing `val1` and `val2` will only result in a 1 (true) if `val1` and `val2` are both 1 (true), 0 (false) otherwise. ORing `val1` and `val2` will result in a 1 (true) if either `val1` or `val2` is 1 (true), 0 (false) otherwise. So you can have more than one condition within a single IF statement at a time, whereas in Batch you'd need to nest them together in a way that fulfills both true and false outcomes for all conditions.

FOR statements operate way differently in Java to how they do in Batch, in Java they're much more traditional compared to Batch's fairly unique implementation. I'm not gonna explain FOR statements as this post is running long enough.

And Java also adds statements Batch does not have, such as WHILE, DO... WHILE, TRY... CATCH... FINALLY and more.

When it comes to core programming concepts, the ideas are much the same with different syntax and functionality, but Java also adds new concepts to the table compared to Batch. Otherwise, it's a completely differently language, and as such it should be treated as one. While `System.out.println()` prints to the console, it shouldn't be thought of in the way that `echo` does, `System.out` is the entire Stream for standard output, so you can do more than just print to it **hint hint**.

White Light - March 19, 2015, 12:31 am

(Mar 18, 2015 11:18 PM)Llamafaggot Wrote: [\[CITE\]](#) Java also uses binary logic, the use of AND and OR operators can be used to "collapse" two separate binary values down to one, for example, ANDing val1 and val2 will only result in a 1 (true) if val1 and val2 are both 1 (true), 0 (false) otherwise. ORing val1 and val2 will result in a 1 (true) if either val1 or val2 is 1 (true), 0 (false) otherwise. So you can have more than one condition within a single IF statement at a time, whereas in Batch you'd need to nest them together in a way that fulfills both true and false outcomes for all conditions. ACTUALLY batch has this capability. If your boolean variables are simple 1's or 0's (not True or False) you can do this:

```
Code set v1=1
set v2=0
set /a bool_and=v1^&v2
set /a bool_or=v1^|v2
set /a bool_xor=v1^^v2
```

set /a bool_not=!v1Although that may not always save time, it can potentially save you from needing multiple nested IF statements in some situations.

Quote:FOR statements operate way differently in Java to how they do in Batch, in Java they're much more traditional compared to Batch's fairly unique implementation. I'm not gonna explain FOR statements as this post is running long enough.

That's just because FOR in batch is the all powerful "loop-over-a-set-of-data" command, which encompasses ranges, items in arrays, outputs of commands, or entire files.

Quote:And Java also adds statements Batch does not have, such as WHILE, DO... WHILE, TRY... CATCH... FINALLY and more.

Which can be implemented in batch using IF statements and GOTO, as well as using %errorlevel% for error handling.

Quote:When it comes to core programming concepts, the ideas are much the same with different syntax and functionality, but Java also adds new concepts to the table compared to Batch.

Otherwise, it's a completely differently language, and as such it should be treated as one. While System.out.println() prints to the console, it shouldn't be thought of in the way that echo does, System.out is the entire Stream for standard output, so you can do more than just print to it *hint hint*.

^^ Really lays out the entire thing here. Batch and Java are like apples and bananas. Both fruit, both taste good, but you wouldn't want an apple if you needed a banana.

annikk.exe - March 19, 2015, 4:45 am

(Mar 19, 2015 12:31 AM)White Light Wrote: [\[CITE\]](#) ACTUALLY batch has this capability. If your boolean variables are simple 1's or 0's (not True or False) you can do this:

```
Code set v1=1
set v2=0
set /a bool_and=v1^&v2
set /a bool_or=v1^|v2
set /a bool_xor=v1^^v2
```

set /a bool_not=!v1Although that may not always save time, it can potentially save you from needing multiple nested IF statements in some situations.

That's cool.. :> I didn't realise you could do that. Or and Xor in particular could be very useful.

When I want to test multiple booleans using "AND" logic, I do it this way:

```
Code set v1=1
set v2=0
```

```
if "%v1%%v2%" == "11" echo TRUE
```

Llamafaggot - March 19, 2015, 5:26 am

(Mar 19, 2015 12:31 AM)White Light Wrote: [\[CITE\]](#) -snip-
Did not know you could do that with IF statements. Interesting...

I did know that you could emulate WHILE loops within Batch, but from what I've read they do have some pretty big limitations and generally are a little advanced to code if you wish to do it efficiently. Else, you could just use a simple label loop with an IF statement that breaks out of the loop if the condition is no longer met.

White Light - March 19, 2015, 11:18 am

(Mar 19, 2015 05:26 AM)Llamafaggot Wrote: [\[CITE\]](#) Else, you could just use a simple label loop with an IF statement that breaks out of the loop if the condition is no longer met.

That's (logically speaking) literally how while loops work.

```
Code :w1
echo blah
set /a n+=1
if not %n%==10 goto :w1
echo out
pause
```

Nakamoto. - March 19, 2015, 6:23 pm

(Mar 19, 2015 11:18 AM)White Light Wrote: [\[CITE\]](#) That's (logically speaking) literally how while loops work.

```
Code :w1
echo blah
set /a n+=1
if not %n%==10 goto :w1
echo out
pause
```

Greenbeans made one a while back that actually simulated a while loop like

```
Code While %tits%=%tits%
```

```
Your shit here
```