SS64   **CMD**   **Syntax**   [   ] Search   **Links**

Using brackets to group expressions

Brackets can be useful to make complex commands more readable while still evaluating variables for the code block as if the command was on a single line.

```
(command)

(
   command )
```

Example

```
IF EXIST C:\pagefile.sys (
    ECHO pagefile found on C: drive)
```

If the command will fit on a single line, then the brackets can be omitted e.g.

```
IF EXIST data.xls Echo The file was found.
```

vs

```
IF EXIST data.xls (
   Echo The file was  found.
)
```

When using brackets the CMD shell will expand [read] all the variables at the beginning of the code block and use those values (even if the variables value is later changed within the code block).

Code blocks can be used to expand variables in places where they would not otherwise be accepted such as the "tokens" parameter of the FOR command:

```
@echo off
set tknz="tokens=1-2"
(
   for /f %tknz% %%A in ('echo') do (
                echo %%B
                echo %%A
          )
)
```

A GOTO command inside a bracketed code block will break the bracket context and cause errors. A GOTO will also break a For-Do Loop. The interpretation of comments/Labels within a bracketed code block is different/complex, so you may wish to put all comments outside the brackets.

The CMD shell statement does not use any great intelligence when evaluating brackets, so for example the command below will fail:

```
IF EXIST MyFile.txt (ECHO Some(more)Potatoes)
```

This version will work:

```
IF EXIST MyFile.txt (ECHO Some[more]Potatoes)
```

You could also escape the extra brackets like `(ECHO Some^(more^)Potatoes)`

It is worth noting that although brackets are legal in NTFS pathnames, such brackets will be misinterpreted by the command processor.

Testing Numeric values

Do not use brackets or quotes if you are comparing numeric values with an IF command
e.g.
```
IF (2) GEQ (15) echo "bigger"
```
or
```
IF "2" GEQ "15" echo "bigger"
```

Will perform a character comparison and will echo "bigger"
however the commands

```
IF 2 GEQ 15 echo "bigger"
```
or
```
IF (2 GEQ 15) echo "bigger"
```

Will perform a numeric comparison and return the correct result.

This behaviour is opposite to the SET /a command where quotes *are* required.

"Yes, we are the richest major nation. But because so much of our national income is concentrated in relatively few hands, large numbers of Americans are worse off economically than their counterparts in other advanced countries" - Paul Krugman

Related

DelayedExpansion - Force the shell to read variables at the start of every line.
IF - conditional command