



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

### ***Windows 11 Help on 'IF' command***

#### ***C:\>HELP IF***

Performs conditional processing in batch programs.

IF [NOT] ERRORLEVEL number command

IF [NOT] string1==string2 command

IF [NOT] EXIST filename command

NOT Specifies that Windows should carry out the command only if the condition is false.

ERRORLEVEL number Specifies a true condition if the last program run returned an exit code equal to or greater than the number specified.

string1==string2 Specifies a true condition if the specified text strings match.

EXIST filename Specifies a true condition if the specified filename exists.

command Specifies the command to carry out if the condition is met. Command can be followed by ELSE command which will execute the command after the ELSE keyword if the specified condition is FALSE

The ELSE clause must occur on the same line as the command after the IF. For example:

```
IF EXIST filename. (  
    del filename.  
) ELSE (  
    echo filename. missing.  
)
```

The following would NOT work because the del command needs to be terminated by a newline:

```
IF EXIST filename. del filename. ELSE echo filename. missing
```

Nor would the following work, since the ELSE command must be on the same line as the end of the IF command:

```
IF EXIST filename. del filename.  
ELSE echo filename. missing
```

The following would work if you want it all on one line:

```
IF EXIST filename. (del filename.) ELSE echo filename. missing
```

If Command Extensions are enabled IF changes as follows:

```
IF [/I] string1 compare-op string2 command  
IF CMDEXTVERSION number command  
IF DEFINED variable command
```

where compare-op may be one of:

EQU - equal

NEQ - not equal

LSS - less than

LEQ - less than or equal

GTR - greater than

GEQ - greater than or equal

and the /I switch, if specified, says to do case insensitive string compares. The /I switch can also be used on the string1==string2 form of IF. These comparisons are generic, in that if both string1 and string2 are both comprised of all numeric digits, then the strings are converted to numbers and a numeric comparison is performed.

The CMDEXTVERSION conditional works just like ERRORLEVEL, except it is comparing against an internal version number associated with the Command Extensions. The first version is 1. It will be incremented by one when significant enhancements are added to the Command Extensions. CMDEXTVERSION conditional is never true when Command Extensions are disabled.

The DEFINED conditional works just like EXIST except it takes an environment variable name and returns true if the environment variable is defined.

%ERRORLEVEL% will expand into a string representation of the current value of ERRORLEVEL, provided that there is not already an environment variable with the name ERRORLEVEL, in which case you will get its value instead. After running a program, the following illustrates ERRORLEVEL use:

```
goto answer%ERRORLEVEL%  
:  
:answer0  
echo Program had return code 0  
:  
:answer1
```

echo Program had return code 1

You can also use numerical comparisons above:

```
IF %ERRORLEVEL% LEQ 1 goto okay
```

%CMDCMDLINE% will expand into the original command line passed to CMD.EXE prior to any processing by CMD.EXE, provided that there is not already an environment variable with the name CMDCMDLINE, in which case you will get its value instead.

%CMDEXTVERSION% will expand into a string representation of the current value of CMDEXTVERSION, provided that there is not already an environment variable with the name CMDEXTVERSION, in which case you will get its value instead.