Windows Batch Scripting: If/Then Conditionals

- Overview
- Part 1 – Getting Started
- Part 2 – Variables
- Part 3 – Return Codes
- Part 4 – stdin, stdout, stderr
- Part 5 – If/Then Conditionals
- Part 6 – Loops
- Part 7 – Functions
- Part 8 – Parsing Input
- Part 9 – Logging
- Part 10 – Advanced Tricks

Computers are all about 1’s and 0’s, right? So, we need a way to handle when some condition is 1, or else do something different when it’s 0.

The good news is DOS has pretty decent support for if/then/else conditions.

Checking that a File or Folder Exists

```
IF EXIST "temp.txt" ECHO found
```

Or the converse:

```
IF NOT EXIST "temp.txt" ECHO not found
```

Both the true condition and the false condition:

```
IF EXIST "temp.txt" (
    ECHO found
) ELSE (
    ECHO not found
)
```

NOTE: It’s a good idea to always quote both operands (sides) of any IF check. This avoids nasty bugs when a variable doesn’t exist, which causes the the operand to effectively disappear and cause a syntax error.

Checking If A Variable Is Not Set

```
IF "%var%"=="" (SET var=default value)
```

Or

```
IF NOT DEFINED var (SET var=default value)
```

Checking If a Variable Matches a Text String

```
SET var=Hello, World!

IF "%var%"=="Hello, World!" (
    ECHO found
)
```

Or with a case insensitive comparison

```
IF /I "%var%"=="hello, world!" (
    ECHO found
)
```

Artimetic Comparisons

```
SET /A var=1

IF /I "%var%" EQU "1" ECHO equality with 1

IF /I "%var%" NEQ "0" ECHO inequality with 0

IF /I "%var%" GEQ "1" ECHO greater than or equal to 1

IF /I "%var%" LEQ "1" ECHO less than or equal to 1
```

Checking a Return Code

```
IF /I "%ERRORLEVEL%" NEQ "0" (
    ECHO execution failed
)
```