

Batch Scripting

Part 2



Parameter Extensions

- When an argument is used to supply a filename then the following extended syntax can be applied (we are using the variable %1, but this works for any parameter) :
 - %~f1 Expand %1 to a Fully qualified path name - C:\utils\MyFile.txt
 - %~d1 Expand %1 to a Drive letter only - C:
 - %~p1 Expand %1 to a Path only e.g. \utils\ this includes a trailing \ which will be interpreted as an escape character by some commands.
 - %~n1 Expand %1 to a file Name without file extension C:\utils\MyFile or if only a path is present (with no trailing backslash\) - the last folder in that path.
 - %~x1 Expand %1 to a file extension only - .txt
 - %~s1 Change the meaning of f, n, s and x to reference the Short 8.3 name (if it exists.)
 - %~1 Expand %1 removing any surrounding quotes (")
 - %~a1 Display the file attributes of %1
 - %~t1 Display the date/time of %1
 - %~z1 Display the file size of %1
 - %~\$PATH:1 Search the PATH environment variable and expand %1 to the fully qualified name of the first match found.

Parameter Extensions

- The modifiers above can be combined:
 - %~dp1 Expand %1 to a drive letter and path only
 - %~sp1 Expand %1 to a path shortened to 8.3 characters
 - %~nx2 Expand %2 to a file name and extension only

Exercise Solution

```
@ECHO OFF
if [%3]==[] GOTO INVALIDINPUT
SET text_value=%1
SET output_folder=%2
Set output_file=%3
IF EXIST %output_folder% (
GOTO AFTERDIRECTORY
)
MKDIR %output_folder%
:AFTERDIRECTORY
CD %output_folder%
attrib -r %output_file%
ECHO %text_value%>%output_file%
attrib +r %output_file%
CD ..
GOTO END
:INVALIDINPUT
echo "invalid input"
:END
```

Then in the command window,

```
HelloWorld.bat "Hello World" "OutputFolder" "output.txt"
```

Basic Operations "FOR"

- `FOR /L %%char IN (start,step,end) DO command [command-parameters]`
 - The set is a sequence of numbers from start to end, by step amount. So (1,1,5) would generate the sequence 1 2 3 4 5 and (5,-1,1) would generate the sequence (5 4 3 2 1).
- `FOR /F ["options"] %%char IN ('string') DO command [command-parameters]`
 - `delims=xxx`
 - specifies a delimiter set. This replaces the default delimiter set of space and tab.
 - `tokens=1,2`
 - specifies which tokens from each line are to be passed to the for body for each iteration. This will cause additional variable names to be allocated

Basic Operations “DelayedExpansion”

- When are the variables that we used up till now expanded?
 - At Parsing Time
- DelayedExpansion can be used to expand variables at execution time instead
 - Ex. 1

```
SETLOCAL EnableDelayedExpansion
```

```
Set _var=first
```

```
Set _var=second& Echo %_var% !_var!
```

- You might expect that to produce the output second second but in fact it will output: first second
- Looking at the line in **bold** above, the line is *parsed* and the variable %_var% is expanded to "first"

Then the line is *executed* (changing the value of _var)

finally !_var! is expanded (because it is using delayed expansion) and it now shows the updated value "second"

Basic Operations “DelayedExpansion”

Ex.2

```
@ECHO OFF
```

```
SETLOCAL EnableDelayedExpansion
```

```
set number=%1
```

```
set power=%2
```

```
set answer=1
```

```
FOR /l %%G in (1,1,%power%) Do (
```

```
echo !answer!
```

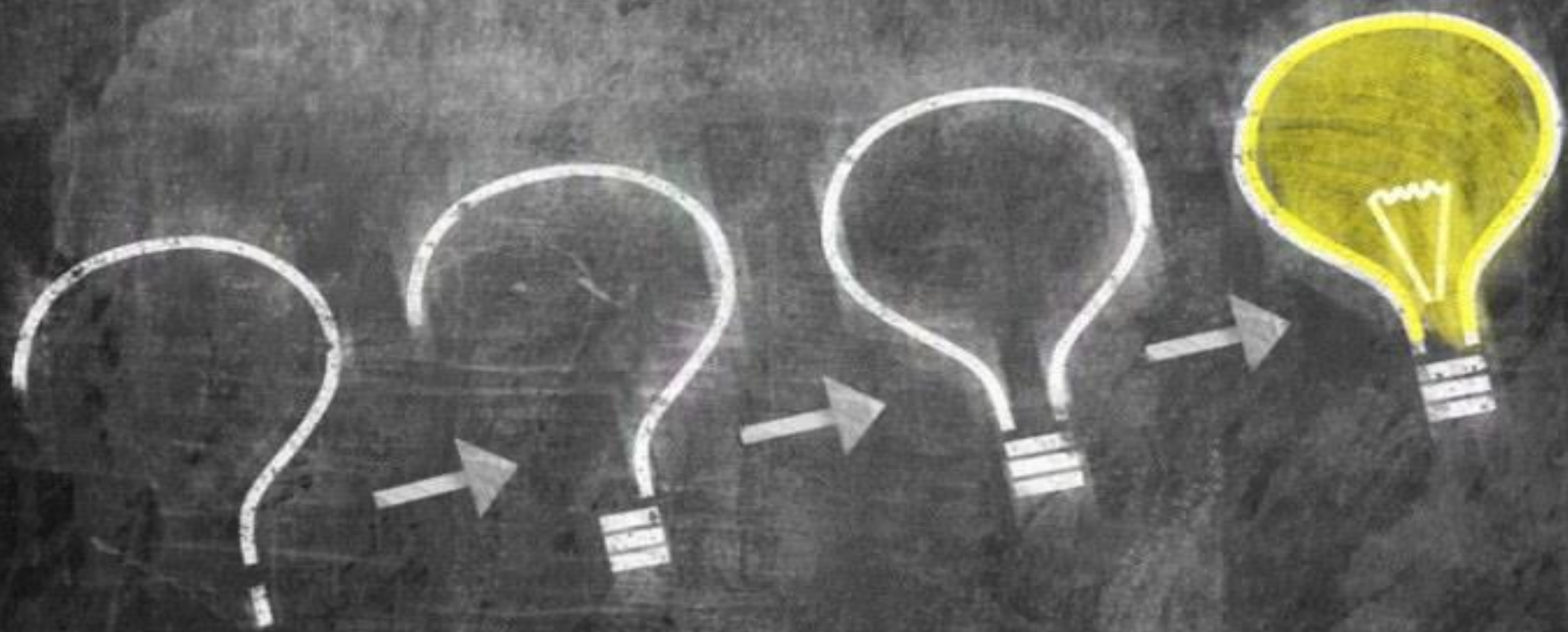
```
set /a answer*=number
```

```
)
```

```
echo answer = %answer%
```

References

- Microsoft Documentation:
<http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds.mspx?mfr=true>
- Command line reference – Web, Database and OS scripting: <http://ss64.com/nt/>
- The Command Line in Windows:
http://www.didiermorandi.fr/vbscript/doc/Windows_Command_Line_Vic_Laurie.pdf
- System Administrator guides (Copy, Xcopy): <http://www.xtraficient.com/batch-files/copy-and-rename-files>
- Redirection reference: <http://www.robvanderwoude.com/redirection.php>
- Shortcut virus removal guide: <http://vintaytime.com/1250/remove-shortcut-virus/>





Eng. Mohammad A.Hekal: embeddedgeek.34@gmail.com

Eng. Omar Soliman: Omar.Soliman@imtSchool.com

Basic Operations "FOR"

```
@ECHO OFF
FOR /F "tokens=1-3" %%G IN (
    "HelloWorld" "OutputFolder" "output.txt"
) DO (
    HelloWorld.bat %%G %%H %%I
)
```

Basic Operations "FOR"

```
@ECHO OFF  
(  
FOR /F "tokens=1-3" %%G IN (  
TXT\input.txt  
) DO (  
HelloWorld.bat %%G %%H %%I  
)  

```

Basic Operations "FOR"

```
@ECHO OFF
```

```
(FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (HelloWorld.bat %%G %%H %%I))>nul  
ECHO "reached end"
```



Self-Study

LOOP THROUGH FILES (RECURSE SUBFOLDERS) - FOR /R

LOOP THROUGH SEVERAL FOLDERS - FOR /D

Basic Operations "CALL"

- Used to call other batch files or exe files within the running batch File

```
@ECHO OFF
```

```
FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
```

```
ECHO "reached end"
```

- Using call returns control to the caller after finishing the command



Self-Study

CALL A SUBROUTINE (FUNCTION CALLS) – [LINK](#)

Basic Operations "COPY"

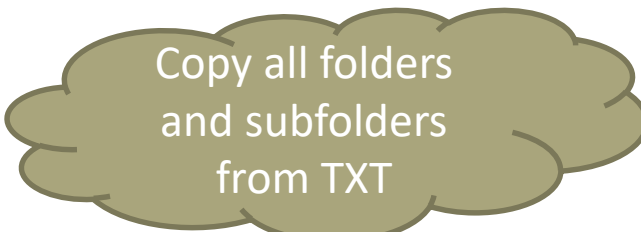
- Copies files from one location to another
 - /Y suppresses prompting if you want to overwrite the file
 - /-Y causes prompting if you want to overwrite the file

```
@ECHO OFF
FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
IF NOT EXIST TXTFiles (MKDIR TXTFiles)
COPY /Y TXT\*.txt TXTFiles
ECHO "reached end"
```

Basic Operations "XCOPY"

- Copies Files/Folders from one location to another with much more options
 - /I assumes that destination is a directory, If destination does not exist and copying more than one file.
 - /R overwrites read-only files
 - /U copies only the files that already exist in the destination
 - /E Copies directories and subdirectories, including empty ones
 - /D:m-d-y Copies files changed on or after the specified date. If no date is given, copies only those files whose source time is newer than the destination time.

```
@ECHO OFF
FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
IF NOT EXIST TXTFiles (MKDIR TXTFiles)
XCOPY /Y /E TXT TXTFiles
ECHO "reached end"
```



Copy all folders
and subfolders
from TXT

Basic Operations "COPY vs XCOPY"

- XCOPY has much more options than COPY
- COPY can do things that XCOPY can't
 - Copy can concatenate files

```
@ECHO OFF
IF NOT EXIST TXTFiles (MKDIR TXTFiles)
COPY /Y TXT\*.txt TXTFiles\collectedInputs.txt
FOR /F "tokens=1-3" %%G IN (TXTFiles\collectedInputs.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
ECHO "reached end"
```

- Copy can copy from the console directly to a file

```
COPY con file.txt
CTRL Z
```

Basic Operations "DEL/RMDIR"

- Deletes Files

```
@ECHO OFF
FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
IF NOT EXIST TXTFiles (MKDIR TXTFiles)
COPY /Y TXT\*.txt TXTFiles
IF NOT EXIST TXTFilesCopy (MKDIR TXTFilesCopy)
COPY TXTFiles TXTFilesCopy
DEL TXTFiles
ECHO "reached end"
```

- To remove a directory use RMDIR

```
RMDIR /S /Q TXTFiles
```

Basic Operations "MOVE"

- Moves Files and Directories

```
@ECHO OFF
FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
IF NOT EXIST TXTFiles (MKDIR TXTFiles)
COPY /Y TXT\*.txt TXTFiles
IF NOT EXIST TXTFilesCopy (MKDIR TXTFilesCopy)
MOVE TXTFiles TXTFilesCopy
ECHO "reached end"
```

Basic Operations "REN"

- Renames Files and Directories

```
@ECHO OFF
FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
IF NOT EXIST TXTFiles (MKDIR TXTFiles)
COPY TXT\*.txt TXTFiles
IF EXIST TXTFilesCopy (RMDIR /S /Q TXTFilesCopy)
REN TXTFiles TXTFilesCopy
ECHO "reached end"
```

"Darkness reigns at the foot of the lighthouse" ~ Japanese Proverb



Final HelloWorld.bat

```
@ECHO OFF
if [%1]==[] GOTO INVALIDINPUT
if [%2]==[] GOTO INVALIDINPUT
if [%3]==[] GOTO INVALIDINPUT
REM get the arguments inside internal variables
SET text_value=%1
SET output_folder=TXT\%2
SET output_file=%3
REM Create output directory if it does not already exist
IF EXIST %output_folder% ( GOTO AFTERDIRECTORY )
MKDIR %output_folder%
:AFTERDIRECTORY
CD %output_folder%
REM removing read only attribute from the file to write to
attrib -r %output_file%
REM writing output text in the output file
ECHO %text_value%>%output_file%
REM adding read only attribute from the file to write to
attrib +r %output_file%
CD..
CD..
goto END
:INVALIDINPUT
echo "invalid input"
:END
```


Final ITI.bat

```
@ECHO OFF
FOR /F "tokens=1-3" %%G IN (TXT\input.txt) DO (CALL HelloWorld.bat %%G %%H %%I)
IF NOT EXIST TXTFiles (MKDIR TXTFiles)
COPY /Y TXT\*.txt TXTFiles
IF EXIST TXTFilesCopy (RMDIR /S /Q TXTFilesCopy)
REN TXTFiles TXTFilesCopy
ECHO "reached end"
```

Exercise

1. Make folders with names as the first string on each line inside the input file.
2. Create files with names of the remaining strings within the same line.
3. Write "0" inside the files.
4. If the folders exist do not overwrite them.
5. If the Files exist do not overwrite them but increment the number inside the file.
6. Make a copy of all ".txt" files into a folder named "Text Files".
7. Make a copy of the ".c" files into a folder "Source Files".
8. Make a copy of the ".h" files into a folder "Header Files".

- Input: Input.txt

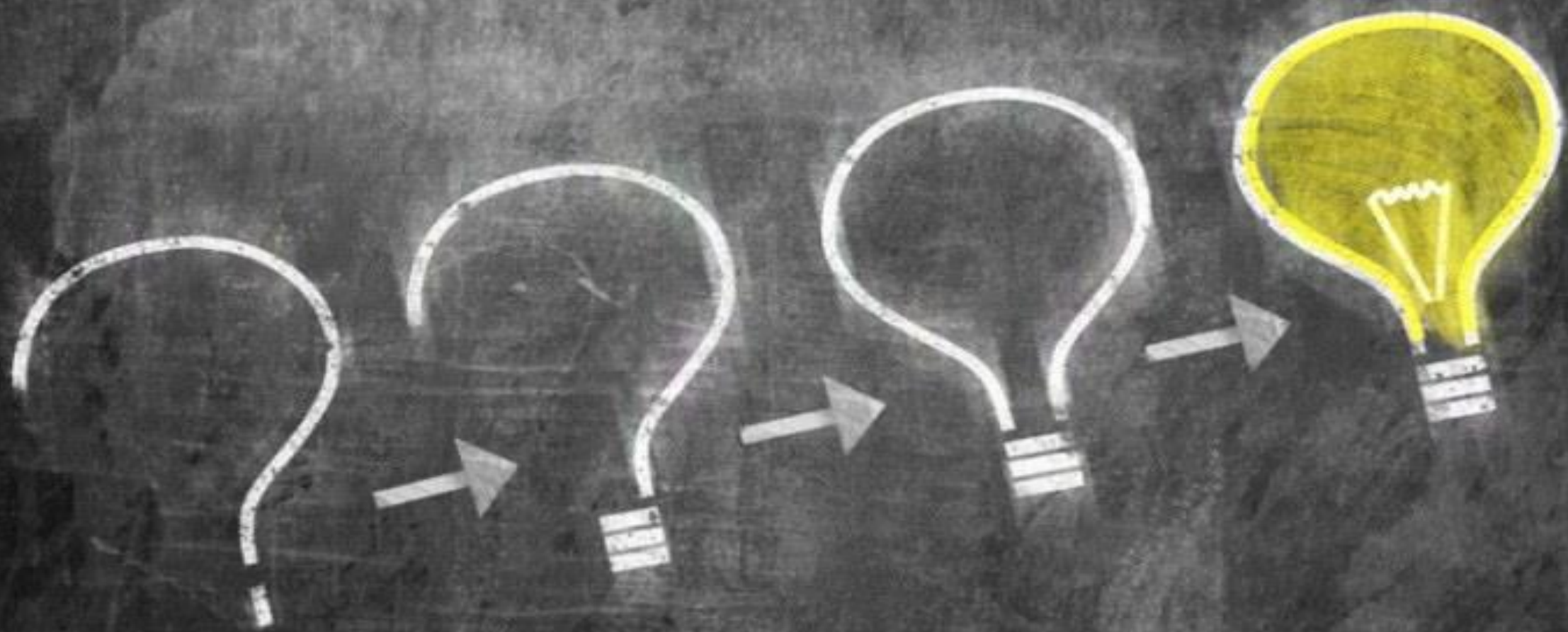
```
Folder1 text1.txt source1.c header1.h
```

```
Folder2 text2.txt source2.c header2.h
```

```
Folder3 text3.txt source3.c header3.h
```

References

- Microsoft Documentation:
<http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds.mspx?mfr=true>
- Command line reference – Web, Database and OS scripting: <http://ss64.com/nt/>
- The Command Line in Windows:
http://www.didiermorandi.fr/vbscript/doc/Windows_Command_Line_Vic_Laurie.pdf
- System Administrator guides (Copy, Xcopy): <http://www.xtraficient.com/batch-files/copy-and-rename-files>
- Redirection reference: <http://www.robvanderwoude.com/redirection.php>
- Shortcut virus removal guide: <http://vintaytime.com/1250/remove-shortcut-virus/>





Eng. Mohammad A.Hekal: embeddedgeek.34@gmail.com

Eng. Omar Soliman: Omar.Soliman@imtSchool.com