Batch Programming
Part I

gameloft

Version

Date	Author	Version	Changelog
20/12/07	Diego.Mercado@gameloft.com	1.0.0	Initial Version
21/12/07	Diego.Mercado@gameloft.com	1.0.1	Added susbt, minor corrections and more examples (some provided by Simon.Carbajal@gameloft.com and Gaspar.Deelias@gameloft.com)
26/12/07	Diego.Mercado@gameloft.com	1.0.2	Fixed an xcopy example, minor stuff, some spell mistakes and the MS url
27/12/07	Diego.Mercado@gameloft.com	1.0.3	Fixed explanation for delayed expansion (thanks Guillermo.Delgadino@gameloft.com). Added examples for getting a substring of a variable
17/04/08	Diego.Mercado@gameloft.com	1.0.4	Fixed bug expanding the TEXT variable for obtaining a substring (thanks fanhieu.kon@gameloft.com)

Guideline

- Basic
- Environment
- Flow control
- File system
- Bibliography

Basic

Basic

help and /?

- Syntax:
  - `HELP [command]`
  - `[command] /?`

- Both are the same. Shows a help for a XP command:
  - i.e.
    - `dir /?`
    - `help set`

Basic

rem, “::” and “;”

- “;” is only used in `config.sys` file
- rem and :: seems that are the same but:
  - At least in MS-DOS rem is slower because `Command.com` parse it

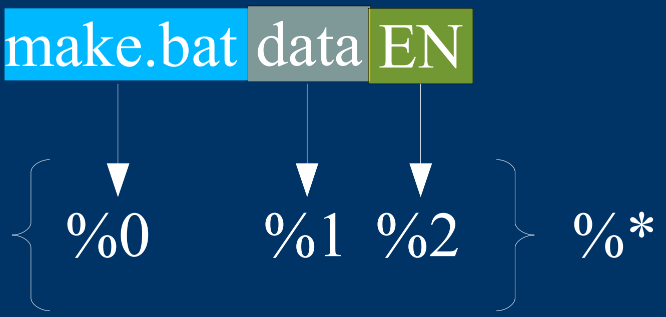
```
rem this is valid comment
:: this is a valid comment too
```

Basic
echo

- Syntax:
  - `echo [{on|off}] [message]`
- On/off activate/deactivate the echo
- @ prevents the echo of the command echo
  - i.e. `@echo blablabla....` prints “blablabla....” only
- Append a line feed
  - i.e. `@echo. >> manifest.mf`
- Use ^ as escaping character
  - i.e. `echo ^> a test`

Basic

getting arguments & shift



- Shift
  - Changes the values of the batch parameters %0 through %9 by copying each parameter into the previous one

```
Shift
%0 data
%1 EN
%2
%* data EN
```

```
Shift /1
%0 make.bat
%1 EN
%2
%* data EN
```

```
Shift /2
%0 make.bat
%1 data
%2
%* data EN
```

Basic
error handling

- When a program stops, it returns an exit code. You can always ask it through the `%ERRORLEVEL%` variable (0 means OK)

```
"%JAVA_HOME%\bin\java.exe" *.java
if not %ERRORLEVEL%==0 (
   goto error
)
goto end

:error
@echo FAILED! Errorcode: %ERRORLEVEL%

:end
```

Basic
error handling

- You can use the binary operators too:
  - `CMD1 && CMD2`
    - Execute CMD2 if and only if CMD1 succeeds.
  - `CMD1 || CMD2`
    - Execute CMD2 if and only if CMD1 fails

```
D:\scripts>cd bogus_dir && cd ..
The system cannot find the path specified.
```

```
D:\scripts>cd bogus_dir || echo %CD%\bogus_dir  missing.
D:\scripts\bogus_dir is missing.
```

Basic

error handling

- Always check if the variable/path exists!!

```
...
rmdir %WORK_DATA_STRING_PATH%\ /s /q > NUL
...
```

1 vs 100 example

THIS WOULD DELETE FROM C:\
IF WORK_DATA_STRING_PATH DOESN'T EXISTS

```
if defined %WORK_DATA_STRING_PATH% (
   rmdir %WORK_DATA_STRING_PATH%\ /s /q > NUL
)
```

Basic

Redirection

Operator	Description
>	Writes the command output to a file or a device, such as a printer, instead of the Command Prompt window.
<	Reads the command input from a file, instead of reading input from the keyboard.
>>	Appends the command output to the end of a file without deleting the information that is already in the file.
>&	Writes the output from one handle to the input of another handle.
<&	Reads the input from one handle and writes it to the output of another handle.
|	Reads the output from one command and writes it to the input of another command. Also known as a pipe.

Basic

Redirection

Handle	Number	Description
STDIN	0	Keyboard input
STDOUT	1	Output to the Command Prompt window
STDERR	2	Error output to the Command Prompt window
UNDEFINED	3 – 9	These handles are defined individually by the application and are specific to each tool.

File	Description
CON	The console
NUL	Void
LPT1	Printer on the 1st parallel port
PRN	Default printer

Basic

Redirection

- Create/override `output.log` file
  - i.e. ipconfig.exe `> output.log`
- Append/create to `output.log` file the ipconfig's STDOUT
  - i.e. ipconfig.exe `>> output.log`
- The same but log also redirect the STDERR
  - i.e. `ipconfig.exe >> output.log 2>&1`
- Redirect STDERR to Void (nothing)
  - i.e. `ipconfig.exe 2> NUL`
  - i.e. `cd bug_dir 2> NUL`

Basic
Redirection

- Sort names.txt and redirect the output to names2.txt
  - i.e. `sort < names.txt > names2.txt`
- Sort (reverse mode) the output of dir
  - i.e. `dir | sort /R`

Environment

Environment
set

- Displays, sets, or removes environment variables. Used without parameters, set displays the current environment settings.

```
set [[/a [expression]] [/p [variable=]] string]
```

`/a` : Sets string to a numerical expression that is evaluated.
`/p` : Sets the value of variable to a line of input.
`variable` : Specifies the variable you want to set or modify.
`string` : Specifies the string you want to associate with the specified variable.

Environment
set

- Substitution:
  - `%VARIABLE:str1=str2%`
- Substring:
  - `%VARIABLE:~start=noChars%`
- Expression
  - Character escape: ^

Operator	**Operation performed**
< >	Grouping
* / % + -	Arithmetic
<< >>	Logical shift
&	Bitwise AND
^	Bitwise exclusive OR
|	Bitwise OR
*= /= = %= += -= &= ^= '|'= <<= >>=	Assignment
,	Expression separator

Environment

set

- Set from the command line “anything”
  - `set /p anything=`
- Removes “anything”
  - `set anything=`
- Set “anything” using an expression
  - `set /a anything=anything + 1`
  - `set /a anything=0x0E`
  - `set /a anything=a^&a`

Environment

set

- If `TEXT==ABC` then...
  - `%TEXT:~0,1%` returns A
  - `%TEXT:~0,2%` returns AB
  - `%TEXT:~1,2%` returns BC
  - `%TEXT:~-3,1%` returns C
  - `%TEXT:~-2,1%` returns B
  - `%TEXT:~-1,1%` returns A
- Substitutes c: by d: in “PATH”
  - `set PATH=%PATH:c:=d:%`

Environment

set

APPDATA	PROCESSOR_IDENTIFIER
CD	PROCESSOR_LEVEL
CMDCMDLINE	PROCESSOR_REVISION
CMDEXTVERSION	PROGRAMFILES
COMPUTERNAME	PROMPT
COMSPEC	RANDOM
DATE	SYSTEMDRIVE
ERRORLEVEL	SYSTEMROOT
HOMEDRIVE	TEMP
HOMEPATH	TMP
HOMESHARE	TIME
NUMBER_OF_PROCESSORS	USERDOMAIN
OS	USERNAME
PATH	USERPROFILE
PATHEXT	WINDIR

Environment
set

- Start -> run -> cmd
  - Users & System variables inhirited:

```
C:\>set
ALLUSERSPROFILE=C:\Documents and Settings\All Users
APPDATA=C:\Documents and Settings\diego.mercado\Datos de
programa
APR_ICONV_PATH=C:\Archivos de programa\Subversion\iconv
CLIENTNAME=Console
CommonProgramFiles=C:\Archivos de programa\Archivos
comunes
COMPUTERNAME=CORWKS0170
ComSpec=C:\WINDOWS\system32\cmd.exe
DEFAULT_CA_NR=CA6
FP_NO_HOST_CHECK=NO
HOME=C:\HOME
HOMEDRIVE=C:
…….
USERDNSDOMAIN=GAMELOFT.ORG
USERDOMAIN=GAMELOFT
USERNAME=diego.mercado
USERPROFILE=C:\Documents and Settings\diego.mercado
windir=C:\WINDOWS
```

Environment
set

- If....

```
C:\>set zz=TEST
C:\>start cmd
```

$\longrightarrow$ This opens another command shell

- What would happen if we call `set` ?

Environment

set

```
C:\>set
ALLUSERSPROFILE=C:\Documents and Settings\All Users
APPDATA=C:\Documents and Settings\diego.mercado\Datos de programa
APR_ICONV_PATH=C:\Archivos de programa\Subversion\iconv
CLIENTNAME=Console
CommonProgramFiles=C:\Archivos de programa\Archivos comunes
COMPUTERNAME=CORWKS0170
ComSpec=C:\WINDOWS\system32\cmd.exe
DEFAULT_CA_NR=CA6
FP_NO_HOST_CHECK=NO
HOME=C:\HOME
HOMEDRIVE=C:
.......
USERDNSDOMAIN=GAMELOFT.ORG
USERDOMAIN=GAMELOFT
USERNAME=diego.mercado
USERPROFILE=C:\Documents and Settings\diego.mercado
windir=C:\WINDOWS
zz=TEST
```

Means just a copy

Every environment is inherited

Environment
setlocal & endlocal

- **SETLOCAL**: Starts localization of environment variables in a batch file. Localization continues until a matching `endlocal` command is encountered or the end of the batch file is reached.

```
setlocal
{enableextensions | disableextensions}
{enabledelayedexpansion |disabledelayedexpansion}
```

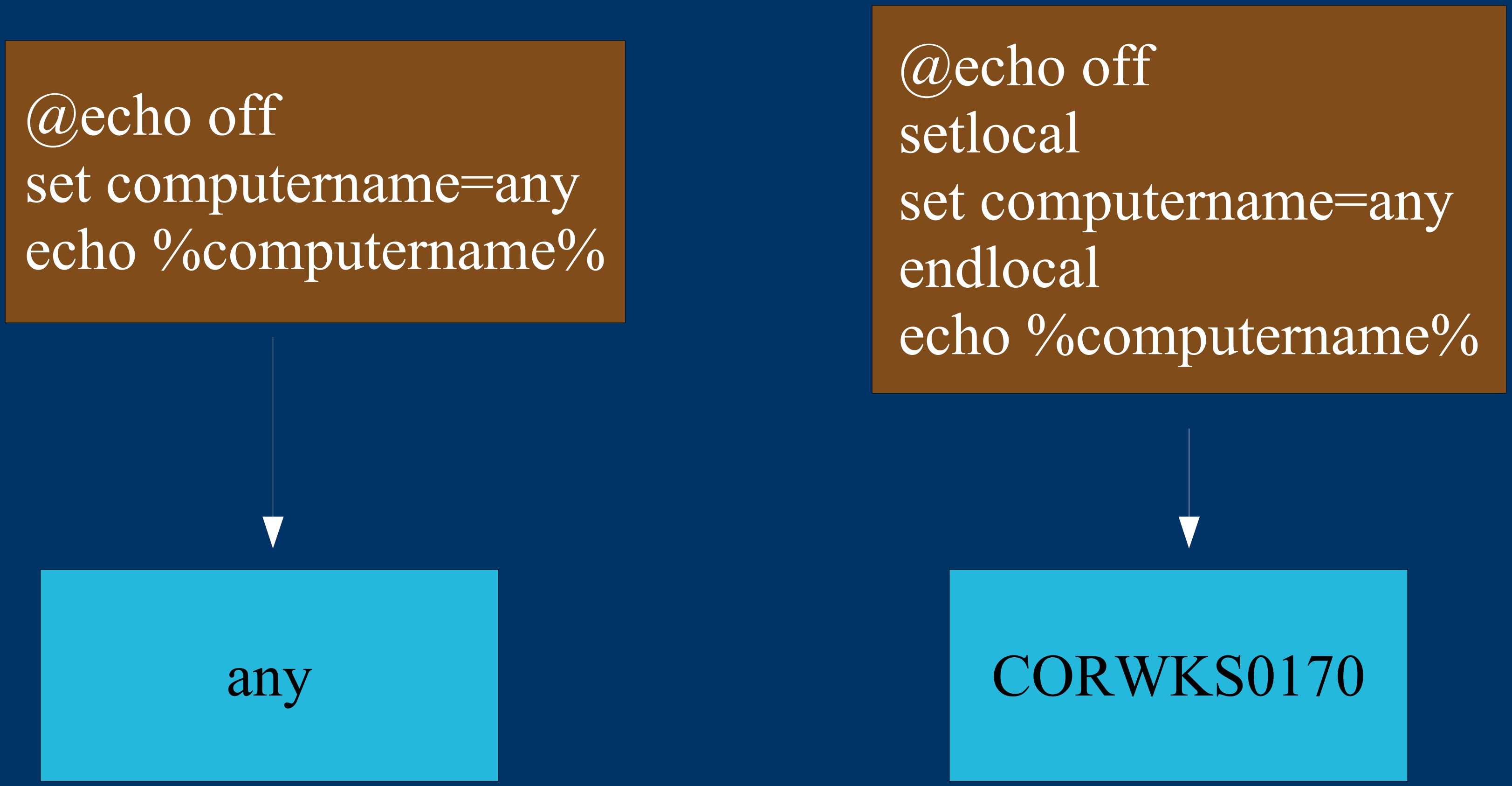
- **ENDLOCAL**: Ends localization of environment changes in a batch file, restoring environment variables to their values before the matching setlocal command

```
endlocal
```

By default: `enableextension disabledelayedexpansion`

Environment

setlocal & endlocal



```
@echo off
set computername=any
echo %computername%
```

```
any
```

```
@echo off
setlocal
set computername=any
endlocal
echo %computername%
```

```
CORWKS0170
```

Environment

setlocal & endlocal

- What are the extensions ?
  - Means extra features
  - Disabling it would remove or change the following commands:
    - DEL o ERASE, COLOR, CD o CHDIR, MD o MKDIR, PROMPT, PUSHD, POPD, SET, SETLOCAL, ENDLOCAL, IF, FOR, CALL, SHIFT, GOTO, START, ASSOC, FTYPE

Environment

setlocal & endlocal

- What is a delayed expansion ?

```
@echo off
set VAR=before
if "%VAR%" == "before" (
   set VAR=after
   if  "%VAR%" == "after" (
     @echo If you see this, it worked
   )
)
```

← Why it doesn't work ?

Environment
setlocal & endlocal

```
@echo off
setlocal enabledelayedexpansion
set VAR=before
if "%VAR%" == "before" (
   set VAR=after
   if "!VAR!" == "after" (
      @echo If you see this, it worked
   )
)
endlocal
```

Delayed environment variable expansion allows you to expand the variable inside of the IF sentence by using a different character (the exclamation mark)

Note: You could set it in the registry or by invoking cmd:

```
c:\cmd /V:{ON|OFF}
```

Flow control

Flow control
if

- `if [not] errorlevel number command [else expression]`
  - i.e. IF NOT ERRORLEVEL 1 goto end
- `if [not] string1==string2 command [else expression]`
  - literal strings or batch variables
  - i.e. IF "game over"==%1 @echo success...

Flow control
if

- `if [not] exist FileName command [else expression]`
  - i.e. IF EXIST c:\autoexec.bat del c:\autoexec.bat
- `if defined variable command [else expression]`
  - i.e. IF DEFINED %TMP%\ rmdir %TMP%\ /s /q

Flow control

if

- `if [/i] string1 CompareOp string2 command [else expression]`
    - /i : Forces string comparisons to ignore case.

Operator	Description
EQU	equal to
NEQ	not equal to
LSS	less than
LEQ	less than or equal to
GTR	greater than
GEQ	greater than or equal to

Flow control

if

- Case/insensitive comparision
  - `IF /i "H" EQU "h" @echo succeed`
  - `IF "h" EQU "h" @echo succeed`
- Using operators
  - `IF %errorlevel% LSS 1 goto okay`

Flow control

call

- Syntax
  - `call [[Drive:][Path] FileName [BatchParameters]] [:label [arguments]]`
- Do not use pipes and redirection
- i.e. `call make.bat %1 %2`

Flow control

call

- Used as a subroutine:

```
@echo off
echo 1
:sub1
echo 2
call :sub2
echo 4
goto end
:sub2
echo 3
goto :EOF
:end
echo 5
```

flow.bat

→ sub1

→ sub2

→ end

What does this ?

Flow control
call

- Used as a subroutine:

```
@echo off
echo 1
:sub1
echo 2
call :sub2
echo 4
goto end
:sub2
echo 3
goto :EOF
:end
echo 5
```

flow.bat

→ sub1

→ sub2

→ end

```
C:\flow.bat
1
2
3
4
5
```

Flow control
pushd & popd

- PUSHD: Stores the actual directory and changes to the specified directory

```
PUSHD [path | ...]

  path   Directory to be changed
```

- POPD: Changes the current directory to the directory stored by the pushd command.

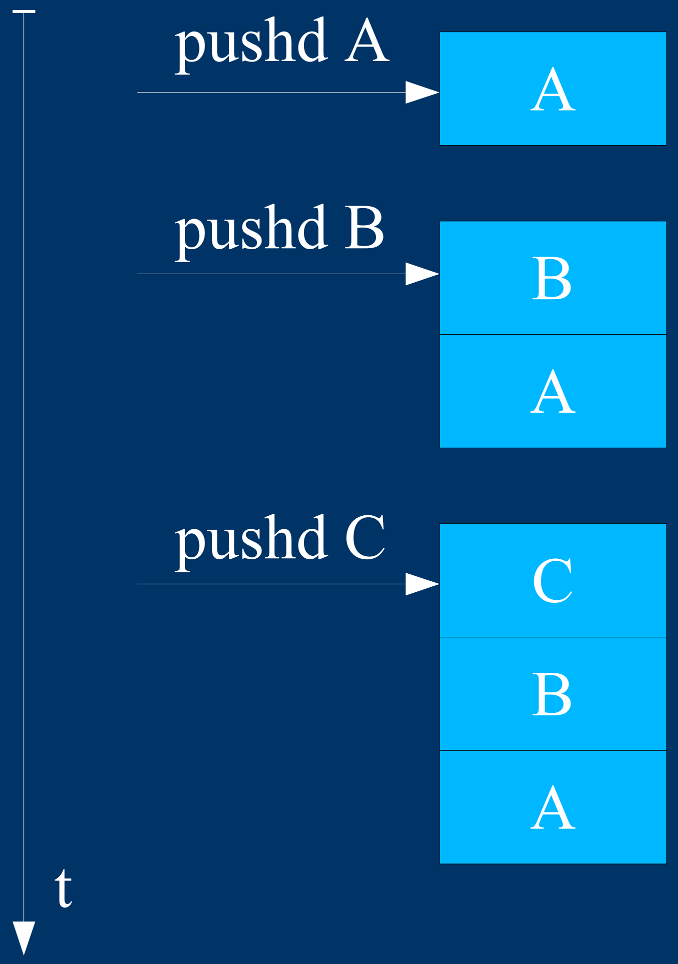
```
POPD
```

Flow control
pushd & popd

- You can store multiple directories by using the pushd command multiple times
- The directories are stored sequentially in a **virtual stack**.

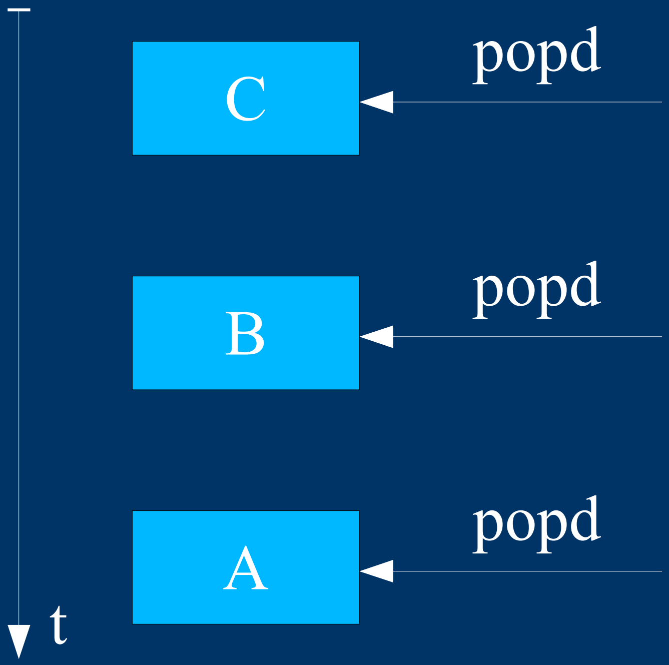
Flow control

pushd & popd



Flow control

pushd & popd



Flow control
pushd & popd

- Example:

```
@echo off
rem This batch file list the *.txt files in a specified
  directory
pushd %1
dir
popd
cls
```

Flow control

pushd & popd

- If command extensions are enabled, the pushd command accepts either a **network path** or a local drive letter and path.
  - Assign temporally to the first unused drive letter. At the end it removes it

Flow control

pushd & popd

- Example:

```
@echo off
rem This batch file list the directories located at a
  network direction
pushd \\Corwks5000\dfs\cor
dir
popd
```

Flow control
for

```
for {%variable|%%variable} in (set) do
  command [ CommandLineOptions]
```

- `{%variable|%%variable}`
  - Both are case-sensitive
  - %% is for scripting and % is for command-line
- `(set)`
  - Specifies one or more files, directories, range of values, or text strings
- `command & CommandLineOptions`
  - The command to be executed with the current args

Flow control

for

- Basically:
  - Take a set of data
  - Make a FOR Parameter %%X equal to some part of that data
  - Perform a command
    - optionally using the parameter as part of the command.
  - Repeat for each item of data

Flow control

for

- syntax-FOR-Files
  - `FOR %%parameter IN (set) DO command`

From command-line

```
c:\>FOR %%F IN (c:\windows\*.*) DO @echo %%F
```

From script

```
FOR %%F IN (c:\windows\*.*) DO @echo %%F
```

Why the fist thing doesn't work?
What does the second one ?

Flow control

for

```
c:\>FOR %A IN (1 2) DO FOR %B IN (A B) DO ECHO %A%B
```

What does this ?

Flow control

for

```
c:\>FOR %A IN (1 2) DO FOR %B IN (A B) DO ECHO %A%B
```

```
1A
1B
2A
2B
```

(Cartesian product)

Flow control

for

- syntax-FOR-Files-Rooted at Path
  - `FOR /R [[drive:]path] %%parameter IN (set) DO command`

From script

```
FOR /R %%F IN (c:\windows\*.*) DO @echo %%F
```

What does this ?

Flow control
for

- syntax-FOR-Folders
  - `FOR /D %%parameter IN (folder_set) DO command`

What does this ?

```
setlocal enabledelayedexpansion
for /D %%i in (*.*) do (
        cd %%i
        if exist make.bat (
                call make.bat %1 %2 %3 %4
                if not %ERRORLEVEL%==0 (
                        echo ERROR:: error in %%i
                        goto error
                )
        )
        cd ..
)
:error
endlocal
```

Flow control

for

- syntax-FOR-List of numbers
  - `FOR /L %%parameter IN (start,step,end) DO command`

```
FOR /L %%G IN (1,1,5) DO echo %%G
1
2
3
4
5
```

Flow control

for

```
FOR /L %%G IN (5,-1,1) DO echo %%G
FOR /L %%G IN (20,-5,10) DO echo %%G
```

What sequences generates?

Flow control
for

```
FOR /L %%G IN (5,-1,1) DO echo %%G
5
4
3
2
1
```

```
FOR /L %%G IN (20,-5,10) DO echo %%G
20
15
10
```

Flow control
for

- syntax-FOR-File contents
  - `FOR /F ["options"] %%parameter IN (filenameset) DO command`
  - `FOR /F ["options"] %%parameter IN ("Text string to process") DO command`
- syntax-FOR-Command Results
  - `FOR /F ["options"] %%parameter IN ('command to process') DO command`

Flow control

for

Keyword	**Description**
eol=c	Specifies an end of line character (just one character).
skip=n	Specifies the number of lines to skip at the beginning of the file.
delims=xxx	Specifies a delimiter set. This replaces the default delimiter set of space and tab.
tokens=x,y,m-n	each iteration. As a result, additional variable names are allocated. The m-n form is a range, specifying the mth through the nth tokens. If the last character in the tokens= string is an asterisk (*), an additional variable is allocated and receives the remaining text on the line after the last token that is parsed.
usebackq	Specifies that you can use quotation marks to quote file names in filenameset, a back quoted string is executed as a command, and a single quoted string is a literal string command.

Flow control

for

```
1.2.3;
```

version.txt

```
@echo off
FOR /F "eol=;tokens=1,2,3 delims=." %%i IN (version.txt) DO (
echo version1 is %%i
echo version2 is %%j
echo version3 is %%k
)
@echo on
```

getVersion.bat

```
C:\>getVersion.bat
version1 is 1
version2 is 2
version3 is 3
```

Flow control
for

- Variable substitution (for any variable I)

Variable	Description
%~I	Expands %I which removes any surrounding quotation marks ("").
%~fI	Expands %I to a fully qualified path name.
%~dI	Expands %I to a drive letter only.
%~pI	Expands %I to a path only.
%~nI	Expands %I to a file name only.
%~xI	Expands %I to a file extension only.
%~sI	Expands path to contain short names only.
%~aI	Expands %I to the file attributes of file.
%~tI	Expands %I to the date and time of file.
%~zI	Expands %I to the size of file.
%~$PATH:I	Searches the directories listed in the PATH environment variable and expands %I to the fully qualified name of the first one found. If the environment variable name is not defined or the file is not found by the search, this modifier expands to the empty string.

Flow control

for

- For instance, this can lead to the following combination:
  - `%~dpI` = drive + path
  - `%~nxI` = file + extension
  - `%~fsI` = full qualified path name + short name
  - `%~dp$PATH:I` = Searches the directories listed in the PATH environment variable for %I and expands to the drive letter and path of the first one found.
  - `%~ftzaI` = full qualified path name + date & time + size + file attributes

Flow control

for

```
FOR %F IN (c:\windows\*.*)
    DO @echo Name:%~nxF - Size:%~zF
....
Name:ADFUUD.SYS - Size:12634
Name:AdfuUpdate.inf - Size:1562
Name:Alcmtr.exe - Size:69632
....
```

Flow control

for

```
for /F %%I in ("%BUILD_RELEASE_PATH%\%JAR_NAME%.jar")
 do set JAR_SIZE=%%~zI echo MIDlet-Jar-Size:%JAR_SIZE%
 >>"%BUILD_RELEASE_PATH%\%JAR_NAME%.jad"
```

What does this ?

File system

File system
del / erase

- Syntax:
  - `{del|erase} [Drive:][Path] FileName [ ...] [/p] [/f] [/s] [/q] [/a[:attributes]]`
    - `/p`: Prompts you for confirmation before deleting the specified file.
    - `/f`: Forces deletion of read-only files.
    - `/s`: Deletes specified files from the current directory and all subdirectories. Displays the names of the files as they are being deleted.
    - `/q`: Specifies quiet mode. You are not prompted for delete confirmation.
    - `/a`: Deletes files based on specified attributes.

File system
mkdir

- Syntax
  - `mkdir [Drive:]Path`
  - i.e. `mkdir Work\0-preprocess\tmp` creates the 3 directories

File system
rmdir / rd

- Syntax:
  - `{rmdir|rd} [Drive:]Path [/s] [/q]`
    - `/s`: Removes the specified directory and all subdirectories including any files
    - `/q`: quiet mode. Without confirmation

File system
xcopy

- Syntax
  - `xcopy Source [Destination] [/w] [/p] [/c] [/v] [/q] [/f] [/l] [/g] [/d[:mm-dd-yyyy]] [/u] [/i] [/s [/e]] [/t] [/k] [/r] [/h] [{/a|/m}] [/n] [/o] [/x] [/exclude:file1[+[file2]][+[file3]] [{/y|/-y}] [/z]`
    - `/s`: Copies directories and subdirectories, unless they are empty
    - `/e`: Copies all subdirectories, even if they are empty
    - `/y`: Suppresses prompting to confirm that you want to overwrite

File system

xcopy

- `/i`: if destination does not exist, xcopy assumes destination specifies a directory name and creates a new directory
- `/f`: Displays source and destination file names while copying
- `/q`: Suppresses the display of xcopy messages
- `/exclude:file1[+[file2]][+[file3]] :` Specifies a list of files containing strings
  - Matchs any part of the string

File system

xcopy

- Copy everything from the local path to D:\Mydir excluding anything that matches the string “obj” (*.obj, \obj\, object, etc...), specified in the file exclude.lst
  - i.e. `XCOPY *.* D:\Mydir /EXCLUDE:exclude.lst`
- Override masters with specific sources (excluding .svn files):
  - i.e. `xcopy /s /a /Y "%SRC_SPEC%" "%SRC0%" > NUL`

Bibliography

- **WINDOWS XP COMMANDS**
  http://www.ss64.com/nt/
- **LENGUAJE DE COMANDOS**
  http://multingles.net/docs/jmt/comandos/comandos1.html
- **Gameloft wiki**
  https://wiki.gameloft.org/twiki/bin/view/Main/BatchScripting
- **Microsoft Command-line reference A-Z**
  http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds.mspx