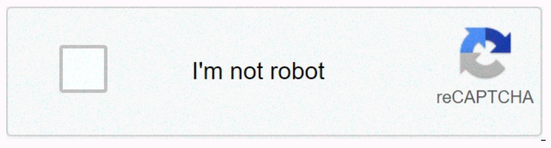

Continue

Windows batch scripting tutorial pdf

Group scripts are stored in simple text files that contain rows with commands executed in section, one at a time. Scripts are a way in which one can reduce these needs by automating the sequence of these instructions to make a person's life on the shell easier and more productive. This tutorial discusses the basic functions of The Batch Script along with relevant examples for a simple understanding. Viewers This tutorial has been prepared for beginners to understand the basic concepts of Batch Script. Prerequisites Knowledge that is reasonable for computer programming and concepts such as variables, instructions, sinks, etc. desired. In Windows, group files are files that store commands in serial order. Command-line interpreters take files as input and perform in the same order. Group files are just text files that are stored in .bat file extensions. It can be written using Notepad or any other text editor. A simple batch file will ECHO OFF ECHO GeeksforGeeks PAUSE After storing it by .bat extension. Double-click it to run the file. It prints the show In the script above, the ECHO off clears the console by hiding commands from printing at the prompt, the ECHO prints the Text GeeksforGeeks to the screen, and then waits for the user to press the key so that the program can be cleaned. Some basic commands of batch files resonate – Prints a string of inputs. It can be Turned on or OFF, for the ECHO to turn the feature on or off. If the ECHO IS INDUCed, the command prompt will display the command is executed. cls – Clear the command prompt screen. Title: Changes the title text displayed above the prompt window. EXIT – To exit Command Prompt. pause - Used to stop execution of Windows group files. :: - Add comments in group files. COPY - Copy batch file or file types in Windows (*.ini) – Initial files. This sets the default variables for systems and programs. CFG (*.cfg) – This is a configuration file. SYS (*.sys) – System files, sometimes editable, are mostly compiled machine codes in new versions. COM (*.com) – Command file. This is an lakuable file for all DOS instructions. In the initial version there are separate files for each command. Now, most are COMMAND.COM. CMD (*.cmd) – These are group files used in the NT operating system. Let's take another example, Let's say we need to list all the files/directory names in a specific directory and save it to a text file, so the batch script for it, @echo off Rem Listing all the files in the Program directory file dir C:\File Program > C:\geeks_list.txt echo Complete! Now when we run this group script, it will create the name of the geeks_list.txt in C:\ directory, displaying all file/folder names in C:\Other useful group Script Program files can be written to diagnose your network and check its performance. :: This batch file check for connection problems. ECHO OFF :: See connection details of ipconfig /all /all circuit Check if geeksforgeeks.com can be reached ping geeksforgeeks.com :: Run the trakeroute to check the route to geeksforgeeks.com tracing the geeksforgeeks.com PAUSE This script is displayed, this script provides information about the current network and some network packet information. 'ipconfig/all' helps view network information and ping & 'tracert' to get every packet information. Learn about ping and traceroute here. Recommended Posts:If you like GeeksforGeeks and want to contribute, you can also write an article using contribute.geeksforgeeks.org or send your article to contribute@geeksforgeeks.org. See your article appearing on the homepage of GeeksforGeeks and help geeks.Please Improve this article if you find anything wrong by clicking on the Article Improve button below. Improved By : shubham_singh This book explains and shows how to use the command translator supplied with Microsoft cmd.exe and related commands, and how to write a Windows batch script for an interpreter. cmd.exe is the default interpreter on all Windows NT-based operating systems, including Windows XP, Windows 7, and Windows 10. Introduction[edit] This book handles the Windows 32-bit command that applies to modern versions of Windows based on the Windows NT environment. It does not address commands specific to DOS environments and DOS-based operating systems, such as Windows 95, Windows 98, and Windows Me, whose Microsoft-supplied command interpreters are actually DOS programs, not Win32 programs. You can find out which version of Windows you run using the VER command. This book first describes using the Windows NT command interpreter, how it receives, describes, and processes commands from users. Then it describes the various instructions available. To get a widespread list of Windows commands and in short summary, open the command prompt on any Windows computer, and type help. To learn about a specific command, type the command name followed by /?. The subject of the book is also known as cluster programming, although clusters refer not only to group files for MS DOS and windows interpreter commands. Other subject terms include group file programming, group file scripts, Windows group commands, Windows group files, Windows command lines, Windows command prompts, and Windows shell scripting. Using a Windows command interpreter,[edit] How the command line is interpreted[edit] The parsing of command lines into command sequences is complex, and varies from command translator to command translator. However, there are four main components: Variable replacement of the scanned command line for variable specifications, and anything found replaced by the contents of those variables. Citing Special characters can be plucked, eliminate their special meaning. The Syntax Command Line is developed into order sequence according to syntax. Redevelopment specifications are used, and removed from before individual commands in sequence are executed. Variable replacement[edit] Command Line can contain variable specifications. This consists of a character % followed by a name, followed by a second % character unless the name is digits in 0 ... 9 or asterisk *. Variable specifications are replaced by the following value: %varname%, such as %PATH% or %USERNAME%, replaced by a named environment variable value. For example, %PATH% is replaced by a PATH environment variable value. %n for 0 & lt;= n &= 9, like %0 or %9, replaced by the n-th parameter value submitted to the group file when it is invoiced, subject to any subsequent modifications by the SHIFT command. For example: %2 is replaced by the second batch file parameter value. %* is replaced by the value of all command line parameters except %0, even if it is outside index 9. The SHIFT command does not affect the results %*. See also command-line arguments. Special names[edit] Some variable names are invisible using set commands. Instead, they are set up to read using % notation. To learn about them, type a help set. Special variable names and what they expand to: Name Replacement Value Used %CD% Current Directory, do not end in slash characters if they are not the current driver root directory %TIME% System time in HH:MM:SS.mm. %DATE% System date in a format specifically for distribution. %RANDOM% pseudo-random number generated between 0 and 32767. %ERRORLEVEL% Error level is returned by the last command executed, or by the last called group script. %CMDEXTVERSION% Number of Command Processor Connection versions currently in use by cmd.exe. %CMDCMDLINE% content of the command line that is used when the current cmd.exe is started. Links: Citing and esting[editing] You can prevent special characters that control command syntheses from having their special meanings as follows, except for percent marks (%): You can circle a string containing special characters with quotation marks. You can put care (^), the character escapes, immediately before special characters. In the command located after the pipe (|), you need to use three care (^^^) for this to work. Special characters that require quotes or escape are usually <, >, |, && and ^. In some instances, ! and \ may need to be offended. A new line can escape using caret as well. When you circle a string using quotation marks, they become part of the argument submitted to the invoiced instructions. On the other hand, when you use caret as an escape character, caret isn't part of the approved argument. Percent mark (%) is a special case. On the command line, it does not require a say or escape unless two of them are used to show variables, such as %OS%. But in group files, you need to use a two percent mark (%%) to generate a single percent sign (%). Includes a percent quotation mark or ahead with doesn't work. Examples of echo Johnson & son echo complete strings instead of separating command lines on &; Character. Favoured quotes also echo Johnson ^& child As above, but use caret before special ampersand characters. No quotes are favoured. echo Johnson & Son Does not use an escape character and therefore, the child is interpreted as a separate command, usually leading to an error message that the command's son cannot be found. echo A ^^ B Echoes A ^ B. Caret needs to escape as well or else it is interpreted as an escape from space. echo > NUL | echo A ^^^^ B Echoes A ^ B. When after the tap, the care used to escape should be three times as many as working; Fourth care is the one who is escaping. if 1 equ 1 ^echo Equal &^echo Indeed, the same Echoes two strings. Caret at the end of the line escaped the new line, leading to three lines treated as if they were a line. Space before first care is required or another 1 will be accompanied by the following echo to produce 1echo. attrib File^ 1.txt Not show a file attribute named File 1.txt because escaping space is not working. Using quotes, as in file 1.txt attrib, works. echo Ratio is 47%. If running from a cluster, the percent sign is ignored. echo Ratio is 47%. If running from a cluster, the percent sign is output once. set /modulo=14%%3 If running from a group, set the modulo variable to 2, the balance divides 14 by 3. Not working with one %. for %%i in (1,2,3) performed echo %%i If running from the group, output 1, 2 and 3. echo %temp% Output temp variable content even if it runs from a group file. The use of percentage marks in groups to access environmental variables and passing arguments does not have to escape. echo ^%temp^% Output literal %temp% when running from the command line. echo %%temp%% literal output %temp% when running batch. online echo / / comment | findstr \/Command FINDSTR uses backslash (\) to escape. Unlike caret, this is internal to the instructions and is unknown to the command shell. Link: Syntax[edit] Command line developed into order sequence according to syntax. In that syntax, simple instructions can be combined to form a pipeline, which may be combined to form a compound command, which can eventually be the instructions connected into the connected order. Simple commands are simply command names, tail commands, and some redeveilment specifications. Examples of simple commands are dir *.txt > somehow. Pipelines are some simple commands combine along with pipe metacharacter—|, also known as vertical bars. Standard output command easily leads each vertical bar connected to input command is easy following it, through pipes. The command interpreter runs all the simple instructions in the pipeline in parallel. Example of the pipeline (consisting of two simple commands) is dir *.txt | More. The compound order is a set of pipelines separated by the in conjunction. Pipelines are implemented sediently, one at a time and joint control over whether the command interpreter performs the next pipeline or not. Examples of compound commands (consisting of two pipelines, which they themselves are only simple commands) are transferring files.txt.bak &; dir > files.txt. In conjunction with: & - In conjunction without function. The next pipeline is always implemented after the current one has finished performing. && - In conjunction with positive conditions. The next pipeline is implemented if the current one is finished executing with zero exit status. || - In conjunction with negative conditions. The next pipeline is implemented if the current one is finished performing with non-zero exit status. The commandments that are crashed brackets are compound orders included in brackets (i.e. (and)). From a syntax standpoint, this turns the compound order into a simple order, whose overall output can be redirected. For example: Command lines (minus temp & dir & popd) > somehow cause the standard output of the entire compound command (minus temp & popd) to be redirected to some. Link: Redirecting[edit] Redirection specifications are used, and removed from the command line, before individual commands in sequence are executed. Control removal specifications where standard input, standard output, and standard error file holder for simple command points. They overcome any impact on file handles that may have resulted from pipelining. (See the previous section on the command syntax.) Signs > and >> can be precasted with 1 for standard output (similar to no prefixes) or 2 for standard errors. The redeveilment specifications are: < file names move standard inputs to read from the named file. > removes the standard output to write to the named file, overwrite the previous content. >> Filename Redirected standard output to write to the named file, adding to the end of the content > previously. &Redirected from handling h. Example: dir *.txt >listing.log Order output dir for listing.log dir file *listing .txt >.log As above; the space before the file name did not make a difference. However, if you type this into the command window, auto-ready with tabs after typing > l actually works, while it doesn't work with >listing.log. dir *.txt 2>NUL Order errors are dir everywhere. dir *.txt >>list.log Order output of the dir command to list.log, added to the file. Therefore, the contents of the file before the redirected command is executed is not lost. dir *.txt >listing.log 2>&1 Instructing command releases to list.log, along with error messages. dir *.txt >listing.log 2>listing .log Code command output is dired to .log listing, and error messages to list errors.log files. >myfile.txt echo Hello Redistribution can get ahead of order. Order. Hello && echo World >myfile.txt The 2nd echo will be redirected. (echo Hello &echo World) >myfile.txt output of both echoes will be redirected. type >myfile.txt Redirects console input (con) to the file. So, allowing multi-row user input to be terminated by users pressing Control + Z. View also #User account. (for %i in (1,2,3) @echo %i) > myfile.txt Redirect the entire output loop to the file. for %i in (1,2,3) perform @echo %i > myfile.txt Start redirecting every time the loop body is entered, lose all output but the latest loop auction. Link: Redirculation ss64.com Using an command redirculation operator in Microsoft How the command is executed[edit] (...) Group reload,[edit] Command interpreter reloads group content after each row or group execution curfew. If you start the following group and convert echo A to echo B in batch shortly after starting it, the output will be B. @echo off the ping -n 6 127.0.0.1 >nul & REM wait for the echo A What is on one non-essential row; change echo A in the following group after walking it has no effect: @echo off ping -n 6 127.0.0.1 >nul & echo A Nor has changes after starting to have any effect on orders knocked down with (and). Therefore, changing the echo A after starting the following group has no effect: @echo off to / L %%i in (1,1,10) do (ping -n 2 127.0.0.1 >nul & REM wait for echo A) Ditto for any other enclosing, including this one: @echo off (ping -n 6 127.0.0.1 >nul & REM wait for echo A) Environmental variables[edit] The environmental variables of the command interpreter process are inherited by any (external) process of the order it executes. Some environment variables are used by the command interpreter itself. Changing its operations. Environment variables are affected by SET, ROUTE, and PROMPT commands. To disset the variable, set it to an empty string, such as a myvar set=. The command interpreter inherits his initial set of environment variables from the process that created them. In case the command interpreter invoiced from this desktop shortcut will be Windows Explorer, for example. The command translator generally has a text user interface, not the graphics, and therefore does not recognize Windows messages informing the application that the environment variable template in Registry has been changed. Changing the environment variables in the Control Panel will cause Windows Explorer to update its own environment variables from the template in Registry, and therefore change the environment variables by which any command interpreter then invoiced will inherit. However, it won't cause command interpreters already running to update environmental variables from the template in Registry. COMSPEC[edit] COMSPEC environment variable contains the full name of the command interpreter program file. This is only inherited from the master process, and is indirectly obtained from the designation of COMSPEC in the variable template in Registry. PATH[edit] The value of the PATH environment variable includes a list of directory names, separated by semi-colon characters. This is a list of sought-after, organized directories, when searching for external command program files to perform. PATHEXT[edit] The variable value of the PATHEXT environment consists of a list of filename extensions, separated by semi-colon characters. This is a list of file name extensions used, in order, when searching for external command program files to perform. The sample content of PATHEXT is printed by echo %PATHEXT%: .COM;. EXE;. BAT;. CMD;. VBS;. VBE;. JS;. JSE;. WSF;. WSH;. MSC By adding .PL to the variable, you can make sure the Perl program runs from the command line even if typed without a .pl. Therefore, instead of typing mydiff.pl a.txt b.txt, you can type mydiff a.txt b.txt. Add .PL to a variable in Windows Vista and then: set PATHEXT %PATHEXT%;. PL If you use the set available in Windows XP, the effect will be temporary and only affects the current console or process. Link: PROMPT[edit] PROMPT environment variables control text published when command interpreters display prompts. Command interpreters display prompts when inducing new command lines in interactive mode, or when selecting group file rows in group file mode. The range of special character sequences in the variable value of the PROMPT environment causes a variety of special effects when prompts are displayed, as in the following table: The $$Yield Expansion Character itself $A & AKA ampersand symbol. Facilities, because it is difficult to put literally & in the PROMPT environment variable value using the SET command. $B vertical bar '|' (pipe symbol) $C Bracket left bracket '(' $D Current date $E ESC (ASCII code 27) $F Senior parents ')' $G Greater-than symbol 'symbol'. >' $H Backspace (delete previous character) $L Less than the symbol '<' $m= remote= name= linked= to= the= current= drive= if= is= a= network= drive;= empty= string= otherwise.= $n= current= drive= letter= $p= current= drive= letter= and= full= path= $q= '=' (equals sign) $S' '= (space= character)= $t= current= system= time= $v= windows= version= number= $_=>>>><CR> (transport return character, aka enter) $+ Because many plus signs (+) because there are items on the directory stack link that are rejected: prompts on the prom ss64 in Microsoft Switches[edit] Most Windows instructions provide the AKA switch option to direct their behavior. Observation: The switch most often consists of one letter; some switches consist of multi-letter sequences. The switch is old-lived with slash (/) rather than, as in some other operating systems, with push marks (-). The switch is insensitive rather than, as in some other operating systems, sensitive cases. If commands from other operating systems are copied to the (such as grade), it usually maintains the preferred convention from the original operating system, including the use of push marks </CR> & case sensitivity. Example: dir/? Displays help. This option is provided by many commands. dir/b/s Lists all files and folders in the current folder repeatedly. Two switches are used: b and s. dir/bs Not working; the switch cannot be grouped behind a single splash. findstr/ric/id: *[0-9]* Files.txt Unlike many other commands, findstr allows the collection of switches behind a single slash. Indeed, r, i and c are one-letter switches. dir/b/s Work. In dir, removing the white space between the command and the first switch or between the switch makes no difference; therefore, is the same as dir/b/s. tree /f/ a Not working, unlike trees/f/a. In trees, separation by white space is mandatory. Did not find /i/v work. dir/od Switch letter or further modified by a letter statering that the order should be by date. Letter d is not a switch on its own. Similar cases include dir/advertisement and many/t4. dir/B/S Switch is insensitive cases, unlike in some other operating systems. compile /r file.txt Organize files in reverse order. sorting/rewind files.txt type allows the switch string to be longer than one letter. compiling/revealing files.txt name allows for a string of switches specified as substring of the full long name of the switch. Therefore, do the same thing as above. stack/reva file.txt not work, because the reva is not a reverse substring. taskkill/im AcroRd32.exe Taskkill requires the name of a multiletter switch for /im; shorten to /i does not work. java -java's version, which comes from the family environment of other operating systems, uses a push convention for its AKA switch options. grade -help If the GNU grade is installed, it requires a multi-letter switch to be served by two d alternates. Error level[edit] Command usually sets the error level at the end of its implementation. In Windows NT and then, it's a 32-bit signed integer; in MS DOS, it used to be an integer from 0 to 255. Keywords: code back, exit code, exit status. Conventional meaning of error level: 0 - success is not 0 - failure The level of error set is usually positive. If the command does not distinguish different types of failures, the level of error on failure is usually 1. Error level usage: It can be tested using &; and ||; see also #Syntax. It can be tested using IF. The value is accessible from the ERRORLEVEL variable. Example: dir >NUL && echo Success Parts after &200 executed only if the error level is zero. color 00 || echo Failure Part after || executed only if the level of error is not zero, either positive or negative. color 00 || (echo failure) Work on multi-line braces as well. echo %ERRORLEVEL% Displays the error level without changing it. if %errorlevel% equ 0 echo Error level is zero, meaning success. if %errorlevel% neq 0 Non-zero error levels, meaning failure. if errorlevel 1 echo Error level is >= 1, meaning a failure through a positive error level. Does not protect failure through negative error levels. Negative. part >=: this does not equal if %errorlevel%saman 1. exit /b 1 Returns a set file, assigning the correctr level to 1. cmd/c exit/b 10 In the middle of a batch file or on the command line, set the correction stage to 10. (cmd /c exit /b 0 & Echo Success) & (cmd/c exit /b -1 || Echo failure) As above, shows the correcting stage is indeed under control. (cmd /c exit /b 0 & cmd /c exit /b 1) || Echo Failure Stage correction chain created by & is the last stage of the chain's referral correction. cmd/c exit/b -1 & otherwise error level 1 echo Will triumph Test if not errorlevel 1, which may appear to test its success, pass at negative number: it tests not correct stage >= 1, which is the correct stage <= 0. set myerrorlevel=%errorlevel% Remember correct level for later. set errorlevel=0 To avoid: overshadowed by deep built-in correcting. Ensures subsequent achievements through %ERRORLEVEL% return 0 instead of the actual correct level. cmd/ c exit / b 0if 1 equ 1 (cmd / c exit / b 1 & echo %errorlevel%) Exposing 0, because %errorlevel% will develop before cmd / c exit / b 1 can be implemented. Link: Processing sequence[edit] Gets non-empty doming substring: setkan a =abcdefgh echo %a:~0.1% & brakes rather than index 0, length 1; decision: echo %a:~1.1% & brakes rather than index 1, length 1; decision: b echo %a:~0.2% & brakes rather than index 0, length 2; decision: ab echo %a:~1.2% & brakes rather than index 1, length 2; decision: bc echo %a:~1% & brakes from index 1 to the end; decision: bcdefgh echo %a:~-1% & brakes from index -1 (last char) to the end; decision: h echo %a:~-2% & brakes from index -2 (onwards to end) to end; decision: gh echo %a:~0,-2% & from index 0 to index -2, excl.; decision: abcdef echo %a:~0,-1% & rem from index 0 to index -1, excl.; decision: abcdefg echo %a:~1,-1% & rem from index 1 to index -1, excl.; decision: bcdefg Testing substring containment: otherwise %a:bc=%=%a% echo yes If the changer contains bc as a substring, echo yes. This test is the one that uses the replacement of the sequence, discussed below. This test does not work if the changer contains quotation marks. Test to begin with: if %a:~0.1%=a echo yes & brake If the changer starts with a, echo yes. if %a:~0.2%=ab echo yes & brake If the changer starts with ab, echo yes. Burst replacement: setkan a=abcd & echo %a:c=%& brakes replace c with nothing; decision: abduction set a=abcd & echo %a:c=e% & brakes replace c with e; decision: annulled; set a=abcd & echo %a:*c=% & brakes replace all so that c with nothing; decision: d Brakes Above, asterisk (*) only works on the pattern you're looking for. See also help for the command SET: set /?. Separates the sequence from everywhere , , and ;: [space, comma and comolon:] set myvar=a b;c;d to %%a in (%myvar%) does echo %%a Separate a sequence by a caned comma, assuming the sequence contains quotation marks: @echo off set myvar=a b;c;d set strippedvar=%myvar% :repeat for /f delims=; %%a in (%strippedvar%) echo %%a set prestripped=%%strippedvar%) prestripped=%%strippedvar%) otherwise %prestrippedvar:=%==%prestrippedvar% goto :repeat limitations: The string above processing does not work with parameter variables (%1, %2, ...). Link: Command line argument[edit] Command line argument AKA command parameters AKA command line submitted to group script can be accessed as %1, %2, ..., %9. There are more than nine arguments; to access it, see how to loop over all of them below. Syntax %0 does not refer to the command line argument but rather to the group file name. Testing for either first command line argument has been prepared: if not -%1-=-- echo One argument is provided if -%1-=---- echo One's argument is not provided &; exit / b Loop strongly over all command line arguments using SHIFT (for each command line argument, ...): :argactionstart if -%1-==-- argactionend goto echo %1 & REM Or do anything else with the argument of goto shift argactionstart :argactionend Loud loop over all command line arguments using SHIFT without modifying %1, %2, etc.: contact :argactionstart %* echo Arg one: %1 & REM %1, %2, etc. not modified in exit this location /b:argactionstart if -%1-==-- argactionend goto echo %1 & REM Or do anything else with the transition of goto argactionstart :argactionend out/b Transfer order line arguments to an environmental variable: setlocal EnableDelayedExpansion REM Prevents affecting callers perhaps group REM Without expansion delay, !arg%argno%! used below will not work. argcount set=0 :argactionstart if -%1-==-- argactionend goto/argcount+=1 set arg%argcount%=%1 shift goto argactionstart :argactionend set argno=0 :loopstart/argno+=1 if %argno% gtr %argcount% goto loopend echo !arg%argno%! & REM Or do anything else with goto loopstart arguments :loopend Loopend over all command line arguments, although not a strong one: for %%i in (%*) do (echo %%i) This looks elegant but not strong, the arguments containing wildcards (*, ?). In particular, the above for command replacement arguments containing wildcards (*, ?) with file names matching them, or dropping them if no files match. However, the loop above works as expected as long as the approved arguments do not contain wild cards. Finding the number of command line arguments, in an unsuccessful way: set argcount=0 to %%i in (%*) set /argcount+=1 More, this doesn't work with an argument that contains a wildcard. The maximum possible number of arguments is greater than 4000, as determined empirically on Windows Vista machines. Numbers can be different on Windows XP and Windows 7. In approving arguments to cluster scripts, the characters used for separation of arguments are the following: comma space equals the character tab sign therefore, the following releases the same four arguments: exam.bat test b d.bat a,b,c,d exam.bat a, b, c, d test.bat a;b;c;d test.bat a=b=c=d test.bat a b,c;,=d Yes, even lines with b,c;,=d pass four hujah, because the order of separating characters is considered as Separator. To have space, commas or commas hollow in the value of the argument, you can surpass the value included in the quotation marks. However, quotation marks become part of the value of the argument. To get rid of quotation marks included when referring to an argument in the script, you can use %~<number> described in #Percent tilde. When passing an argument to an invoiced command instead of a cluster script, you usually need to separate the instructions from the first argument using space. However, for internal commands, the separation is not necessary if the first character after the command name is one of several symbols, including .\/, and more: echo. Tree. Failed: tree. not found. trees are external instructions. dir.. Lists the contents of the master directory. cd.. Changing the current directory to parents. cd\ Changes the current directory to the root. Start. Open Windows Explorer from the current directory. dir/b/s List directory content repeatedly, showing the full path. Link: Wildcards[edit] Many commands receive wildcards-character file names that don't stand for themselves and allow matching a group of filenames. Wildcards: * (asterisk): any character sequence? (question marks): single characters other than periods (.) or, if part of a question mark sequence at the end of the maximum period is free of file names, perhaps the number of zero characters; see examples for Example Explanation: dir *.txt Match Myfile.txt, Plan.txt and any other files with .txt connection address. dir *txt Duration does not need to be entered. However, this will also match the named files without a period convention, such as myfiletxt. ren *.cxx *.cpp Rename all files with a .cxx extension to get .cpp extension. dir a?b.txt Match files aab.txt, abb.txt, a0b.txt, etc. Do not match the aerial .txt, since question marks are followed by characters other than question marks or duration cannot match zero characters. Do not match the .b.txt, because the question marks do not match the duration. dir ???. txt Matches .txt, a.txt, aa.txt, and aaa.txt among other things, since each ask mark in sequence followed duration can match the number of zero characters. dir a???. B???. Txt??? Match .b.txt, among other things. Although the last question mark sequence is not followed by a period, it is still a sequence at the end of the maximum period free of file names. dir ????????. &txt; @REM ask the same file as *.txt, because each file also has a short file name that does not have more than 8 characters before .txt. Quirk with a short file name: wildcard matching is done both on a long file name and which is usually hidden short of 8 chars + duration + 3 chars file names. This can lead to bad shocks. Unlike the shells of some other operating systems, the cmd shell.exe not perform wildcards (replacement of patterns containing wildcards with a list of file names matching patterns) by themselves. It is the responsibility of each program to treat & lt;/number> & </number> therefore. This allows things like ren *.txt *.bat, because the compose command actually looks at * wildcards instead of a list of files that match the wildcards. Therefore, the echo *.txt not display files in the current folder that matches the pattern but instead displays *.txt. Another consequence is that you can write findstr a.*txt without fear that the a.*txt section will be replaced by the name of some files in the current folder. Furthermore, the findstr/s recursive pattern *.txt possible, while in some other operating systems, parts of *.txt will be replaced by the filename found in the current folder, ignoring nested folders. Orders that accept wild cards include ATTRIB, COPY, DIR, FINDSTR, FOR, REN, etc. Link: Free card on ss64 Using wildcard characters in Microsoft Users[edit] You can get input from users using the following methods: COMMAND OPTIONS SET/P command Using cone type >myfile.txt, where multi-row user input is terminated by users pressing Control + Z. Tilde Percentage [edit] When the command line argument contains a file name, a special syntax can be used to get a variety of information about the file. The following syntaxes expand to a variety of information about the files approved as %1: Example Syntax Expansion Results %~1 %1 without quotation marks included Not provided %~f1 Full path with drive letter C:\Windows\System32otepad.exe %~d1 Drive Letters C: %~p1 Drive path with backslash trailing \Windows\System32\ %~n1 For files, unseen file names and extensionsFor folders, folder name notepad %~x1 Filename extensions include duration of .exe %~s1 Modify f, n and x to use short name Not provided %~a1 File attributes --a------ %~t1 Date and last modification time of file 02.11.2006 11:45 %~z1 File size 151040 %~pn1 Combined p and n \Windows\System32otepad %~dpnx1 Combined multiple letters C:\Windows\System32otepad.exe %~$PATH: 3 1 The first full matching path found in the folder inside the PATH variable , or an empty string in a match no. %~n0 %~n used for %0:Group connection name tildetest %~nx0 %~nx used for %0:Group name tildetest.bat %~d0 %~f is used on %0:Group C drive letter: %~dp0 %~dp applied to %0: Group folders with backslash trailing C:\Users\Joe Hoe\ The Syntax is applied to a single letter variable created by a FOR command, such as %%i. To learn about this subject from the command line, type call /? or for /?. Link: AKA subprogram functions can be emulated using CALLS, labels, SETLOCAL and ENDLOCAL. Example of function that specifies arithmetic power: @echo call :p 2 4 echo %results% rem Print 16, is determined as 2 * 2 * 2 * 2 goto :eof rem __Function power________________________ Rem Argument: %1 and %2 :p set counter=%2 set interim_product=%1 :p ower_loop %counter% gtr 1 (set /a interim_product=interim_product * %1 set /kaunter=- 1 1 :p ower_loop) endlocal & Set results=%interim_product% goto :eof While goto :eof at the end of the function is not really required, it needs to be there in general cases where there is more than one function. Variables where the results should be saved can be determined on the call line as follows: @echo off call :sayhello results=world echo %results% out/b:sayhello set %1=Hello %2 REM SET %1 to set the value back out/b In the example above, exit/b applied instead of goto :eof to the same effect. Also, remember that the same sign is a way to separate the parameters. Therefore, the following things achieve the same: call :sayhello results=world call:sayhello the result of the world call :sayhello, World calls :sayhello results;world (See Command line argument as a reminder) Link: Calculations[edit] Batch script can perform simple 32-bit integer arithmetic and slight manipulation using /SET The largest supported Integer is 2147483647 = 2 ^ 31 - 1. The smallest supported integer is -2147483648 = - (2 ^ 31), can be allocated with set tricks /num=-2147483647-1. Syntax is reminding C. Arithmetic Operator language including *, /, % (modulo), +, -. In batch, the modulo needs to be included as %%. The Bitwise operator interprets the numbers as a 32-digit sequence of binary digits. This is ~ (complementary), & (and), | (or), ^ (xor), << (left transition), >> (right transition). The logic operator of ignorance is !: it turns zero into one and not zero into zero. The combined operator is ,: it allows for more calculations in a set of commands. The operator of an affiliate assignment is modeled on +=, which, in a +=b, means a=a+b. Therefore, a-=b means a=a-b. The same goes for *=, /=, %=, &=, ^=, |=, <<=, and >>=. The operator's preference orders supported, are as follows: (( ) * / + - << >> & ^ ^ ^ ==/= %= += == ^= ^= <<= >>= , Literally can be included as decimals (1234), hexadecimal (0xffff, lead 0x), and octroll (077), in-house representation of the negative number is complementary This provides a link between arithmetic operation and bit operation. For example, -2147483648 is represented as 0x800000000, and therefore set / num=~(2147483647-1) produces 2147483647, which is equivalent to 0x7FFFFFFF (set type/num=0x7FFFFFFF to check). Since some operators have special meanings for command translators, the phrase that uses them should be included in quotation marks, such as these: prescribed / num=255^127 set /a num=255^127 Alternative placement quotation marks. set / a num=255^^127 Escape ^ use ^ instead of quotation marks. Example: set n1=40 &; set n2=25set /a n3=%n1%n2% Using standard percent notation for variable expansion. set n1=40 &40 set n2=25set /a n3=n1+n2 Avoid percentage notations around variables as not required for / a. set / a num=255^127 Encloses ^ in quotation marks to avoid special meanings for command translators. set / a n1 = (10 + 5)/5 Space around = no matter with / a. However, getting to him lends himself to writing a var set = value without /a, which sets the value of the var instead of the var. if 1==1 (set /a n1=(2+4)*5) Does not work: the arithmetic bracket in the group bracket causes problems. if 1==1 (set /a n1=^(2+4^)*5) Escape arithmetic brackets with caret works (^), as symbolized n1=2+(4*5) in quotation marks. set /a n1=2+3.n2=4*7 Perform two calculations. set /a n1=n2=2 Has the same effect as n1=2,n2=2. set n1=40 &40 set n2=25 &30 set /a n3=n1+n2 set la n1=2,n2=3,n3=n1+n2 set n1=40 && set n2=25 &; set /a n3=%n1%n2% Not working unless n1 and n2 are previously set. Variable specifications %n1% and %n2% will be expanded before the first set command is executed. Dropping percent notation makes it work. set /a n1=2,n2=3,n3=%n1%+n2% Did not work unless n1 and n2 were previously set, for the reasons stated in the previous example. set /a n1=0xffff Set n1 using hexadecimal notation. set /a n1=0777 Set n1 using octane notation. set /a n1=%random% pseudo-random number from 0 to 32767 = 2^15-1. set /a n1=%random%>>10 Pseudo-random numbers from 0 to 31 = 2^5-1. The transition senior operator fell 10 of 15 bits, saving 5 bits. set /a n1=%random%%50 Pseudo-random numbers from 0 to 49. Using modulo operator %. In the group, %% is required for the modulo: set /a n1=%random%50. Because of the use of this modulo, the result is not perfect uniform; it is uniform if the 2nd modulo operand-above 50-equals power 2, for example 256 = 2^8. set /a n1=(%random%<15)+%random% Pseudo-random number from 0 to 1073741823 = 2^30 - 1. Combines two 15-bit random numbers generated by %random% only to generate a single 30-bit random number.. set /a n1=(%random%<15)+%random%)%)%1000000 As above, but again use the modulo, this time to reach ranges from 0 to 99999999. An example calculation that prints prime numbers: @echo off setlocal set n=1 :p rint_primes_loop set /a n=n+1 set of cand_divisor=1 :p rint_primes_loop2 set/a cand_divisor=cand_divisor+1 set /a cand_divisor_squared=cand_divisor*cand_divisor if %cand_divisor_squared% gtr %n% echo Prime %n% & goto :p rint_primes_loop set /a modulo=n%cand_divisor if %modulo% equ 0 goto :p rint_primes_loop & REM Not a prime goto :p rint_primes_loop2 Links: set at ss64.com set at Microsoft Random Numbers at ss64.com Finding files[edit] Files can be found using #DIR, #FOR, #FINDSTR, #FORFILES, and #WHERE. Example: dir/b/s *base*.doc* Output all the files in the current folder and its subfolders so that the filename before the extension contains the basic words and its extension begins with the doc, which includes doc and docx. The file is output with a full path, one file per row. dir/b/s*.txt | findstr/i pers.*doc Combines results from removing files including their complete path with filtering commands that supports limited ordinary expressions, resulting in a versatile and powerful combination of finding files with their names and directory names. Their. /r %i in (*) performs a @if %~zi geq 100000 echo %~zi%~i For each file in the current folder and its subfolders that have a size larger than or equal to 1,000,000 bytes, removing the file size in bytes and full path of files. For syntax in %~zi, see #Percent tilde. forfiles /s/d 06/10/2015 /c cmd/c echo @fdate @path For each file in the current folder and its subfolders are modified on June 10, 2015 or later, output the date of modification of the file and full file path. The date format after/d is locally specific. Therefore, enables to find the most recently modified files. (for /r %i in (*) @echo %~ti :: %i) | findstr 2015.*:: Looking for the current folder repeatedly, the file output for which its last modification date was in 2015. Put the date and time of the modification, followed by a double colon, before the file name. Works as long as the Windows and local versions used display dates in formats that contain four digits of the year. Multiple colons are used to ensure findstr commands match dates instead of file names. to / r %i in (*) @echo %~ti | findstr 2015 >NUL && echo %i As above, the output file changed in 2015. Unlike the above, only file output, not modification date. findstr/ i/s/m cat.*mat *.txt Find files by their content. Perform full text searches for regular expressions of paint.*mat in files with names ending .txt, and output file names. The /m switch ensures that only file names are output. where *.bat Output all .bat numbers in the current directory and in the directory available in the PATH. Keyboard shortcuts[edit] When using Windows command lines from a standard console that appears after typing a cmd.exe after pressing Windows + R, you can use multi-keyboard shortcuts, including function keys: Tabs: Complete the relevant section of the string typed from the file name or folder name in the current folder. The relevant section is usually the last part free of space, but the use of quotation marks changes them. Generally considering both files and folders to complete, but cd commands only assume folders. Up and down arrow keys: Enter commands from the command history, one at a time. Escape: Delete the command line while typed. F1: Character type from a single that was previously entered commands from the command history, one character at a time. Each subsequent F1 newspaper entered another character. F2: Asks you to type characters, and enter the shortest prefixes of previous commands from a command history that doesn't include characters typed. Therefore, if the previous instructions are the echo hello of the world and you type o, enter the ech. F3: Enter the previous single command from the command history. Recurrent pressing has no further effect. F4: Asks you to type characters, and delete parts which starts at the cursor's current location, continues to the right, and ends with the script you entered excluding that script. Therefore, if you click the world echo Hello, place the cursor in H using the left left key, press F4 and then w, you get the echo world. If you press F4 and then Insert, delete the text from the cursor to the end of the row. F5: Enter the previous command from the command history, one at a time. F6: Enter Control+Z. F7 characters: Opens a character-based pop-up window with command history, and allows you to use the arrow keys and enter to select commands. After you press pop-ups, the order is immediately executed. F8: Given the typed string, showing items from the history of the command that have that string as a precursor, one at a time. F9: Allows you to enter the number of commands from the command history, and then execute the order. Alt + F7: Delete the order history. The above is also known as an instant command keyboard shortcut. The availability of the above shortcuts does not seem to depend on running DOSKEY. Link: Windows Keyboard shortcuts ss64.com doskey in Microsoft Paths[edit] File paths and directories follow certain conventions. This includes the use of drive letters that may be followed by a colon (:), the use of backslash (\) as a pass breaker, and the difference between relative and absolute paths. Slash forward (/) often works when used instead of (\) but not always; it is usually used to indicate a switch (optional). Using slash forward can lead to a variety of clear, and best avoidable behaviors. Special device names include NUL, CON, PRN, AUX, COM1, ..., COM9, LPT1, ..., LPT9; this may be redirected. Example: attrib C:\Windows\System32otepad.exe is successful if a file exists, as it should. This is an absolute route with a driving letter. It is also known as a fully qualified route. attrib \Windows\System32otepad.exe Is successful if the current drive is C:, and if the file exists, as it should. This is an absolute route without a driver letter. cd /d C:\Windows &;; attrib System32otepad.exe Succeeded if a file exists. The path given to the attrib is a relative path. cd /d C:\Windows\System32 & attrib C:notepad.exe Succeeded if a file exists. The path given to the attrib is relative even containing a driving letter: there should be a C:otepad.exe with the backslash for it to be an absolute route. cd /d C:\Windows &;; attrib .\System32otepad.exe Succeeded if the file exists. A period marks the current folder. attrib . A period marks the current folder. cd /d C:\Windows &;; attrib .\System32\\otepad.exe Succeeded if the file exists. Piling backslashes has no effect beyond the first backslash. cd /d C:\Windows &;; attrib .\System32 Succeeded if the folder exists. cd /d C:\Windows &;; attrib .\System32\ Failed. Folders are usually screened without a final backslash. cd C:\Windows\System32\ cd .. A double period indicates the parent folder. attrib C:\Windows\System32\.. \.. \Windows\System32 Duration can be used in the middle of the path to navigate to the parent folder, even if multiple times. attrib \\myserver\myvolume UnC Route network starts with multiple backslash and no driver letters. cd \\myserver\myvolume \\myserver\myvolume not working; the server folder in this direct way does not work. rejected \\myserver\folder creates drivers for folders and changes to them automatically. After you use the #POPD, the driver will not be re-signed. attrib C:/Windows/System32/notepad.exe Successfully on various versions of the cmd.exe. Using slashes forward. Link: Arrays can be emulated in delayed expansion mode using a combination of % and ! to indicate a variable. There, %i% is a variable value i with immediate expansion while !i! is the variable value i in the delayed expansion. @echo setlocal EnableDelayedExpansion for /l %%i in (1, 1, 10) do (set array_%i=!random!) for / l %%i in (1, 1, 10) do (echo !array_%i!) : For each item in various, do not know the long set i=1 :startloop if not defined array_%i% set endloop goto array_%i%=array_ i <2> <9>%i%!_dummy_suffix echo A%: !array_%i%! set /i+=1 goto startloop :endloop Links: Perl one-liners[edit] Some tasks can be easily reached with Perl one-liner. Perl is a script language that comes from another operating system environment. Because many Windows computing environments have been installed Perl, perl one-liner is a natural and compact connection of Windows group scripts. Example: echo abcbbc| perl -pe s/a.*?c/ac/ Lets Perl acts as a sed, a utility that supports the replacement of the specified text using a common expression. echo a b | print $F barrels[1] Allow Perl to act as a cut command, display the 2nd field or line column, in this case b. Use $F[2] to display the 3rd field; indexing begins at zero. Original solution: FOR /f. perl -ne print if / \x22hello\x22/file.txt Acts as a grep or FINDSTR, removing the line in the file.txt which matches the usual expression after if. Using common powerful Expressions of Perl, more powerful than FINDSTR. perl -ne$. <= 10 and print MyFile.txt Lets Perl acts as a header command -10, removing the first 10 lines of the file. perl -e sleep 5 for / f %i in ('perl -MPOSIX -le print strftime '%Y-%m-%d', local time) performs @set isodate=%i Get the current date in ISO format into an isodate variable. perl -MWin32::Clipboard -e print Win32::Clipboard->Get() Output text contents of the clipboard. When saved to get .bat, producing a useful getclip command to complete the CLIP instructions. perl -MText::D iff -e print different 'File1.txt', File2.txt Output differences between two files in a format similar to the different commands known from other operating systems, including the context line, the line begins with + and the line begins with -. perl -MWin32::Sound -e Win32::Sound::P lay('C:\WINDOWS\Mediaotify.wav'); Plays the sound of notifications in .wav without showing any windows. On the web, Perl one-liner often in other operating system command line conventions, including the use of apostrophe (') to surround arguments rather than Windows quotation marks. This needs to be tweeted for Windows. Link: Unix command[edit] Windows cmd.exe cmd.exe The interpreter can use instructions from an operating system such as Unix, provided it is installed. Example instructions include growing, sed, awk, wc, head and tail. Commands are available from the GNU project, and their Windows ports exist. You can learn more about the instructions in the Guide to Unix Wikibook. Be careful that group programs that rely on these commands are not guaranteed to work on other Windows machines. Licensed Windows version of the GNU command can be obtained from the following projects: GnuWin32, sourceforge.net ezwinports, sourceforge.net: has some ports fresher than GnuWin32 Alternative ways of running the GNU order for Windows 10 is The Windows Subsystem for Linux. There are no common touch instructions from other operating systems. The touch command will modify the file's last modification timetamp without changing its contents. One workaround, with unclear reliability and usability across multiple versions of Windows, is this: Links: Built-in commands[edit] This command is all built for the translator of the command itself, and cannot be changed. Sometimes this is because they need access to the data structure of the internal command interpreter, or modify the process properties of the command interpreter itself. Overview[edit] Description of ASSOC Command Associates an extension with a file type (FTYPE). Set REST or clear the CTRL+C review extended. CALL Calling a group program from another. CD, CHDIR Displays or sets the current directory. CHCP Displays or sets the active code page number. CLS Clears the screen. COLOR Sets the background color and background of the console. COPY Copy file. DATE Displays and sets the system date. DEL, DELETE one or more files. DIR Displays a list of files and subdirectories in the directory. ECHO Displays a message, or turns on or off commands that resonate or switch off. ELSE Performs conditional processing in batch programs when if it is not true. ENDLOCAL Ends the browsing of environmental changes in group files. EXIT Stop from CMD.EXE program (command translator). TO Run the command specified for each file in a set of files. FTYPE Sets file type commands. GOTO Go to the label. IF Perform conditional processing in batch programs. MD, MKDIR Creating a directory. MOVE Transfer files to a new location PATH set or modify the PAUSED PATH environment Causes the command session to be paused for user input. POPD Changes to drivers and directories appear from the PROM set of directory arrangements or modify the strings displayed while waiting for input. PUSH the current directory to the stack, and change to a new directory. RD/ RMDIR Eliminates directory. REM Comment instructions. Unlike a double colon (::), instructions can be executed. REN/RENAME Renaming files or SET or display shell environment variable SETLOCAL Creates a children's environment for group files. SHIFT Moves the cluster parameters forward. START Starting a program with a variety of options. TIME Displays or sets the SYSTEM Clock Title window title type Prints the contents of the file to the console. VER Shows the command processor, operating system version. VERIFY That a copy of the file has been done correctly. VOL Shows the current volume label. ASSOC(edit] Associates extensions with file types (FTYPE), displays existing associations, or deletes associations. See also FTYPE. Example: assoc Lists all associations, in format <file=>=<file type=>, for example, .pl=Perl or .xls=Excel.Sheet.8. assoc | .doc list all associations that contain substring .doc. Link: BREAK[edit] In a Windows version based on Windows NT, nothing; stored for compatibility with MS DOS. Example: > blank.txt Creates a blank file or to clear the contents of an existing file, take advantage of the fact that fracking is nothing and has no output. Shorter to type than nul > empty .txt. Link: CALL[edit] Calls a group program from another, calls a subprogram in a single group program, or, as an undocumented behavior, starts a program. In particular, suspend the execution of the caller, begin executing the caller, and resume the execution of the caller if and when the caller finishes execution. To call a subprogram, see Functions Section. Be careful that calling the group program from a group without using the results of the call keywords in the execution never returned to the caller once the caller is complete. The caller environment variables, and unless the caller blocks it through SETLOCAL, changes made by the caller to the environment variable become visible to the caller once it continues execution. Example: mybatch.bat If used in groups, transfer controls to mybatch.bat and never resumes caller execution. call mybatch.bat call mybatch call mybatch.bat arg1 arg 2 call :mylabel call :mylabel call :mylabel arg1 arg 2 cmd /c mybatch.bat Just like the call, but continue the execution despite the error. Furthermore, any changes made by environmental variables are not disseminated to callers. notepad .exe Launch Notepad, or in general, any other execution. This appears not to be the use of intended calls, and is not formally documented. See also Functions, CMD amd START. Link: CD[edit] Changes to other directories, or displays the current directory. However, if different drive letters are used, it does not switch to different drives or volumes. Example: cd Current directory output, for example. C:\Windows\System32. cd C:\Program Files No quotes around the route with space. cd \Program Files cd Document cd %USERPROFILE% cd/d C:\Program Files Changes to driver directory C: even C: is not the current driver. C: & Cd Set The Way. Changes to the C: although C: boost directory are not current boosters. Cd.. Changes to the parent directory. There's nothing if it's already in the root directory. Cd.. \.. Changes to the parent directory are two rankings up. C: & cd</file> </file> </file> & Cd.. \.. \Program Files Using .. to navigate the directory tree up and down the cd \\myserver\folder Not working. Converting the directory directly to the Universal Naming Convention (UNC) folder network is not working. Keywords: UNC Password. subst A: \\myserver\folder &; cd/d A: Transform the directory to the server folder using #SUBST, asserts the drive letter A: is free. rejected \\myserver\folder creates drivers for folders and changes to them automatically. After you use the #POPD, the driver will not be re-signed. cd C:\W* Changes to C:\Windows, in normal Windows setup. Therefore, wildcards work. Useful for manual typing from the command line. cd C:\W**32 Changes to C:\Windows\System32, in normal Windows setup. Link: cd ss64.com Microsoft CHDIR[edit] CD synonyms. CLS[edit] Clear screen. COLOR[edit] Sets the background color and console background. Examples: f9 Color Use a white background and a blue background. Color Restores the original color settings. Links: colors on ss64.com on Microsoft COPY[edit] Copy files. See also MOVE, XCOPY and ROBOCOPY. Example: copy F:\File.txt Copying files into the current directory, asserts the current directory is not F:\. copy F:\My File.txt As above; quotation marks are required to surround the file with space. copy F:*.txt Copy files located in F:\ and ends in the dot txt into the current directory, asserts the current directory is not F:\. copy F:*.txt . Is the same as the command above. copy File.txt Error Message, because the .txt can not be copied on its own. copy File1.txt File2.txt Copy Files1.txt to File2.txt, overwrite File2.txt if verified by the user or if running from a batch script. copy File.txt My Directory Copy File.txt into my Directory directory, ass ass as the My Directory exists. A copy of Dir1 Dir2 Copies of all files located directly in the Directory dir1 to Dir2, assuming Dir1 and Dir2 are directories. Did not copy files located the retired directory Dir1. copy *.txt *.bak For each *.txt in the current folder, making copies ends with a bakery instead of txt. Link: copy ss64.com Microsoft DEL[edit] Delete files. Use with caution, especially in combination with wildcards. Just delete a file, not a directory, which sees RD. For more information, type del/?. Example: del File.txt del/s *.txt repeatedly delete files including retired directories, but save directories; compassionately delete all corresponding files without asking for verification. del/p/s *.txt As above, but request confirmation before each file. del/q* .txt Delete without asking for verification. Links: del ss64.com del in Microsoft DIR lists directory content. Offers a wide range of options. Type dir/? for further assistance. Example: dir files and folders in the current folder, excluding hidden files and system files; use different listing methods if the DIRCMD variable is not blank and contains a switch to dir. dir. D: dir /b dir/s Lists directory content and all recurring subdirectories. dir /s/b Lists the contents of the directory and all subdirectories repeatedly, one file per row, displays a complete path for each file or directory listed. dir *.txt Lists all other .txt extensions. dir/a Includes hidden files and system files in the listing. dir/ah dir / advertisement List directory only. Other letters after /A include S, I, R, A and L. dir / ahd List of hidden directories only. dir /a-d List files only, ignoring directories. dir / a-d-h List of non-hidden files only, ignoring the directory. dir/od Order files and folders by last modification date. Other letters after /O include N (by name), E (by extension), S (by size), and G (first folder) dir/o-s Command file with decreasing size; the effect on the folder order is unclear. dir /-c/o-s/a-d List file ordered with descant size, ignoring thousands of separatists through /-C, excluding folders. dir /s/b/odd lists the contents of the directory and all recurring subdirectories, ordering the files in each directory on the last modification date. Messages only occur for each directory; the complete set of files to be found is not ordered in total. dir/a/s List repeatedly includes hidden files and system files. Can be used to know the use of the disk (directory size), considering the final line of output. Link: dir on ss64.com dir on Microsoft DATE[edit] Displaying or setting a date. The way the date is displayed depends on the country settings. The date can also be displayed using echo %DATE%. Getting a date in iso format, such as 2000-01-28: that's a simple place, because the date format depends on the country settings. If you can assume the format Of Monday 01/28/2000, the following will do: isodate set=%date:~10.4%-%date:~4.2%-%date:~7.2% If you have A WMIC, the following is a local free: for /f %i in ('wmic os get LocalDateTime') performs @if %i lss if %i gtr 0 local set=%iset isodate=%localdt:~0.4%-%local:~4%-%local:~2%-%local:~6.2% To use above in the group above, %i into %%i and remove @ from previous if. If you have installed Perl: for /f %i in ('perl -MPOSIX -le print strftime '%Y-%m-%d', local time) performs @set isodate=%i Link: ECHO[edit] Displays the message, or turns on or off the resonating or dead command. Example: echo on @echo off echo Hello echo hello echo %PATH% Displays path variable content. echo Owner ^&son Use caret (^) to escape ampersand (&),therefore allow echoing ampersands. echo 1&echo 2&echo 3 Displays three strings, each followed by a new line. echo. Output of new lines during the non-output period. Without duration, echo off or echo on. Adding space before the period leads to the period being output. Other characters that have the same effect as the period include :;,/\(+[]. echo %random%>>MyRandomNumbers.txt Although it seems to be issuing random numbers to MyRandomNumbers.txt, it doesn't actually do so for numbers 0-9, because of this, when placed before >>;, it doesn't actually do so for numbers 0-9, because of this, when placed before >>;, it doesn't actually do so for numbers 0-9, because of this, when placed before >>;, it doesn't actually do so for numbers 0-9, because of this, when placed before >>;, it doesn't actually do so for numbers 0-9, because of this, when placed before >> channel to which ones to redirect. See also #Redirection. echo 2>>MyRandomNumbers.txt Instead of selecting 2, directing a standard error to the file. (echo 2)>>MyRandomNumbers.txt Echoes even a small number (in this case 2) and direct the results. >>MyRandomNumbers.txt echo 2 Other ways to resonate even a small number and move the results. Displaying a new lineless string requires tricks: <NUL =output= of= a= command:= displays= output= of= a= command:.= output= of the= next= next= will= be= displayed= immediately= after= :.= set=>lt;/NUL> <NUL =current= time==&time= displays= current= time:= followed= by= the= output= of time= .= (set=>>/NUL> <NUL =current= time=&time=)==>set tricks.txt As before, by redirecting Link: echo at ss64.com Microsoft ELSE[edit] Example: if a file exists.txt (echo File exists.) others (echo Files do not exist. See also IF. ENDLOCAL[edit] Ends the set of local environment variables starting using SETLOCAL. Can be used to create subprograms: see Functions. Link: endlocal in ss64.com endlocal at Microsoft ERASE[edit] Synonym DEL. EXIT[edit] Exit the DOS console or, with /b, only groups currently running or subroutine are in progress. If used without/b in a group file, cause the DOS console to call the group to close. Example: Link: exiting ss64.com microsoft FOR[edit] Chewing a series of values, executing commands. In the following example, %i will be used from the command line while %%i will be used from the group. The index (for example, %i) must be a single character variable name. Example: for %%i in (1,2,3) performing echo %%i In batch, resonating 1, 2, and 3. In batch, commands must use a two percent mark. The selus examples are intended to be pasted directly into the command line, so they use a one percent mark and include @to avoid repeated views. for %i in (1,2,3) @echo %i From the command line, resonates 1, 2, and 3. Commands for attempting to interpret items as file names and as filename patterns that contain wildcards. It doesn't complain if the item doesn't match the existing file name, though. for %i in (1,2,a*d*c*c*e*t) performs @echo %i Unless you have files matching the third pattern, resonate 1 and 2, remove the third item. for %i in (1 2,3;4) @echo %i Echoes 1, 2, 3, and 4. Yes, a mixture of item separation is used. for %i in (*.txt) @echo %i Echoes the file name located in the current folder and has .txt extension. for %i in (C:\Windows\system32*.exe) there is no @echo %i Echoes file name that matches the pattern. for /r %i in (*.txt) not @echo %i Echoes file names with full path, files that have extensions to .txt anywhere in the current folder including Nested. for /d %i in (*) @echo %i Echoes the names of all folders in the current folder. for /r /d %i in (*) @echo %i Echoing names including full path of all folders in the current folder, including nested folders. for /r %i</NUL> %i</NUL> (*) is there a @if %~zi geq 1000000 echo %~zi %i For each file in the current folder and its subfolders that have a size greater than or equal to 1,000,000 stanzas, the output size of the file in the temple and the path is full of files. For syntax in %~zi, see #Percent tilde. for /l %i in (1,1,10) @echo %i Echoes numbers from 1 to 10. for /f token=* %i in (list.txt) @echo %i For each line in the file, echoing the line. for / f token=* %i in (list1.txt list2.txt) do @echo %i For each line in the file, echoing the line. for /f token=* %i in (*.txt) @echo %i Nothing. Do not accept wild cards to match the file name. for / f token=1-3 delims=: %a in (First:Second::Third) @echo %c-%b-%a Parses burst into token addressed by :. Quotation marks indicate a sequence not a file name. The second and third tokens are stored in %b and %c even though %b and %c are not mentioned manifestly in the command section before doing so. Both colons are considered to be one separator; %c is not but vice versa Third. Are there some cutting referral jobs from other operating systems. for / f token=1-3* delims=: %a in (First:Second::Third:Fourth:Fifth) does @echo %c-%b-%a: %d As above, only the 4th and 5th items are captured in %d as Fourth:Fifth, including separators. for / f token=1-3* delims=, %a in (First, Second,:Third:Fourth:Fifth) @echo %c-%b-%a: %d Possible registrants. for /f token=1-3 %a in (First Second Third,item) @echo %c-%b-%a Defaulting registrants are spaces and tabs. Thus, they are different from the separators used to separate the arguments passed to the group. for /f token=* %i in ('cd') @echo %i For each command decision line, echoing the line. for / f token=* %i in ('dir /b /a-d-h') performs @echo %~nxai For each file is not hidden in the current folder, exposing the file attribute followed by the file name. In the sequence %~nxai, using syntax described on the tilde #Percent. for / f usebackq tokens=* %i in ('dir /b /a-d-h') performs @echo %~nxai As above, but uses backquote script (') around the command to be executed. for / f token=* %i in ('task list ^| compile ^&echo End') perform @echo %i Paip and ampersands in the command to be executed should be regardless of using caret (^). (for deep %i (1,2,3) do @echo %i) > manaoldtemp.txt To redirect the entire coil result, place the entire coil in the cage before redirecting. Otherwise, the diversion will bind the coil body, so any new melting of the coil body will overcome the previous decision. for %i in (1,2,3) @echo %i > any paper.txt Examples related to the above. He showed the consequences of failing to put a coil in the cage. Go on: To jump to Next loop and thus emulate the known continuous statement from many languages, you can use the goto provided you put the loop body in the subroutine, as indicated in the following: for %%i in (b c) contact :for_body %%i exit :for_body echo 1 %1 goto :cont echo 2 %1:cont exit /b If you use goto hold inside for coil, the use of goto breaks the coil book rack. The following failed: for %%i in (b c) do (echo 1 %i goto :cont echo 2%i:cont echo 3 %i) Link: for ss64.com for Microsoft FTYPE[edit] Expose or assign commands to perform for file types. See also ASSOC. Example: ftype Lists all union commands to be executed with a file type, for example, 'Perl=C:\Perl\bin\perl.exe %1 %*' ftype | search Excel.Sheet A list of only unions whose exposure lines contain Excel Links.Sheet: GOTO[edit] Go to labels. Example: goto :mylabel echo Hello 1 REM Hello 1 was never printed. :mylabel echo Hello 2 goto :eof echo Hello 3 REM Hello 3 was never printed. Eof is a virtual label that stands for the end of the file. Goto in the coil body makes cmd forget the coil, even though the label is in the same coil body. Link: go to ss64.com goto in Microsoft IF[edit] By effectively executing commands. Documentation can be found by entering IF /? to the CMD rush. Available basic exams: exist <filename> <string>==<string> <expression1>sama -- equal to <expression2> <expression1>neq <expression2>-- not equal to lss -- less than leq -- less than or <expression1>same <expression2> <expression1> <expression2> <expression1>gtr -- greater than geq -- greater than or equal <expression2> <expression1> <expression2> <variable>estimated <number>errorlevel cmdextversion For each test <number>asas, should not be used. Apparently there are no controllers such as AND, OR, and others to combine the basic exams. Suis / I made == and saman comparison ignore the case. Example: if not exist %targetpath% (echo target path not encountered. exit /b) Example: otherwise 1 equ 0 echo Not the same if 1 equ 0 echo A & echo B Nothing; both echo commands are conquered to the terms. if not 1 equ 0 goto :mylabel if not geq b echo No bigger if b geq echo Greater if b geq A echo Greater in case-insensitive comparison if B geq echo Greater in insensitive case-case comparison if 0 equal 0 equal 00 echo Numeric equality if not 0==00 echo String inequality if 01 geq 1 echo The comparison of numbers if not 01 geq 1 echo String comparison if 1 equ 0 (echo Equal) other echo Notices are not the same confinement around the positive then partly to make it work. otherwise a==Echo case sensitive inequality if /i a==Case-insensitive equality if /i==/i echo It doesn't work if /i==/i echo Equal, uses quotation marks to avoid the literal meaning of /i Link: if in ss64.com if in Microsoft MD[edit] Create a new directory or referrer. Has a synonymous MKDIR; also rd antonym. Example: md Dir Creates a directory in the current directory. md Dir1 Dir2 Creates two directories in the current directory. md My Dir With Spaces Creates a directory with a name that contains space in the current directory. Link: md in ss64.com md in Microsoft MKDIR</number> </number> </variable> </expression2> </expression1> </expression2> </expression1> </expression2> </expression1> </expression2> </expression1> </expression2> </expression1> </string> </string> </filename> </filename> MD. MKLINK Creates a symlink or other types of links. Available since Windows Vista. Link: copy ss64.com on Microsoft MOVE[edit] Move files or directories between directories, or rename them. See also REN. Example: move File1.txt File2.txt Rename Files1.txt to File2.txt, overwrite File2.txt if verified by the user or if running from a group script. moving Files.txt Transfer files.txt into the Dir directory, assuming Files.txt are files and Dir is a directory; overwrite the target file Dir\a.txt if the conditions for the alternate are met. move Dir1 Dir2

Renamed Directory Dir1 to Dir2, assuming Dir1 is a directory and Dir2 does not exist. move Dir1 Dir2 Mobile directory Dir1 to Dir2, resulting in the existence of Dir2\Dir1, assuming both Dir1 and Dir2 are existing directories. move F:\File.txt Moves the file to the current directory. move F:*.txt Move files located in F:\ and ends in the dot txt into the current directory, asserts the current directory is not F:\. Links: moving ss64.com Microsoft PATH or setting the path environment variable value. When releases, including PATH= at the beginning of the output. Example: Route Output route. Example output: PATH=C:\Windows\system32; C:\Windows; C:\Program File\Python27 path C:\Users\Joe Hoe\Scripts;%path% Extends path with C:\Users\Joe Hoe\Script, only using the cmd process.exe. route ; echo %path% | perl -pe s//g | select Show folders in the sorted path if you are already installed. Link: the route ss64.com route in Microsoft PAUSE[edit] Prompting users and waiting for the input queue to be included. Link: pause on SS64.com the Microsoft POPD Pause[edit] Changes to drivers and directories emerge from directory stacks. Stacks of directories are filled using pushD commands. Links: filled ss64.com on a popd on Microsoft PROMPT[edit] Can be used to change or reset the cmd.exe prompt. It sets the value of the PROMPT environment variable. C:\>PROMPT MyPrompt$G MyPrompt>CD C:\ MyPrompt>PROMPT C:\> Prompt Instructions used to set prompts to MyPrompt>. The CD indicates that the current directory path is C:\. Using PROMPT without any parameters sets prompts back to the directory path. Link: a prompt on the ss64.com prompt at Microsoft PUSHD[edit] Pushes the current directory to a directory heap, making it available for POPD commands to recover, and, if executed with an argument, changes to the directory specified as an argument. Link: rejected on the ss64.com microsoft RD[edit] Remove directory. Take a look at the RMDIR and MD antonyms that are synonymous. Each default, only a blank directory can be removed. Also type rd/?. Example: rd Dir1 rd Dir1 Dir2 rd My Dir With Spaces rd/s Dir1 Removes the Dir1 directory including all files and subdirectories therein, requesting verification once before proceeding Removal. To delete repeated files in a retired directory with the authentication of each file, use DEL with /s switch. rd/q/s Dir1 Dir1 above, but without asking for verification. Link: rd at ss64.com rd in Microsoft REN[edit] Rename files and directories. Example: ren filewithtpyo.txt filewithtypo.txt ren *.cxx *.cpp Links: RENAME[edit] This is a RE-command synonym. REM is used for words in group files, preventing the content of the comment from being executed. Example: REM An unfiltered statement of echo Hello REM This speech will be displayed by echo hello & REM This statement is ignored as required :: This verse has been marked as a comment using a multiple colon. REM is usually placed at the beginning of the line. If placed behind the command, it does not work, unless preceded by the ampersand, as shown in the example above. Multiple colons are an alternative to REM. It can cause problems when used in the middle of a massage in bracke, as used in FOR loops. The colon doubles as if only tricks, a label that begins with the colon. Link: RMDIR[edit] This is synonymous RD. SET[edit] Displaying or setting environment variables. With / P switch, it asks users for input, stores results in variables. With / Switch, it performs simple arithmetic calculations, keeping the results in variables. With string assignments, there is no space before and after signs of equality; therefore, the set of names = Peter doesn't work, while the set of names=Peter does. Example: Set Displays a list of designated HOME environment variables Displays an environment variable value whose name starts with the MYNUMBER HOME set=56 SETS OF HOME=%HOME%; C:\Program Files\My Bin Folders set /P user_input=Enter integer: set /A result = 4 * (6 / 3) Sets the result variable with the calculation result. See also #Calculation. Links: set to ss64.com Microsoft SETLOCAL[edit] When applied in group files, make all further changes to the local environment variable to the current group file. When used outside of group files, nothing. Can be screened using ENDLOCAL. Exit group files automatically calls end locally. Can be used to create subprograms: see Functions. Furthermore, it can be used to enable delayed expansion like this: setlocal EnableDelayedExpansion. The delayed expansion consists of a variable name included in an exam mark replaced by its value only after the implementation reaches its usage location rather than at the previous point. Here's an example of using delayed expansion in scripts that print the first number of file rows specified, provides several known head command functions from other operating systems: @echo off call :myhead 2 File.txt exit/b :: Myhead function :: ==%2 - file name :myhead setlocal EnableDelayedExpansion counter set=1 for/f tokens=* %%i in (%2) (echo %i set / counter=!counter!+1 if !counter! gtr %1 out /b) exit /b Link: SHIFT[edit] Removes group file arguments together, but does not affect %*. Therefore, if %1=Hello 1, %2=Hello 2, and %3=Hello %3=Hello then, after SHIFT, %1=Hello 2, and %2=Hello 3, but %* is Hello 1 Hello 2 Hello 3. Links: ss64.com microsoft START[edit] Start a program in a new window, or open a document. Using an unclear algorithm to determine whether the first approved argument is the title of the window or program to be implemented; Hypothesis: it uses the presence of quotes around the first argument as an indication that it is the title of the window. Example: start a notepad.exe &; echo Complete. Start the notepad.exe, proceeds to the next command without waiting to end the start. Keywords: asynchronous. start notepad.exe Launch a new console window with a notepad.exe its title, apparently an unwanted result. start C:\Program Files\Internet Explorer\iexplore.exe Start Internet Explorer. Empty passed as the first argument is the title of the console window that doesn't actually open, or at least doesn't look so. start C:\Program File\Internet Explorer\iexplore.exe Launching a new console window with C:\Program File\Internet Explorer\iexplore.exe as its title, seems an unwanted result. start / wait for notepad.exe &note; echo Complete. Starting .exe, waiting for it to end before proceeding. start /low notepad.exe &clean; echo Complete. As above, but start a program with low priorities. start MyFile.xls Open documents in the program that is allocated to open it. start Start a new console (command line window) in the same current folder. Start. Opens the current folder in Windows Explorer. start mailto: Starting an app to write a new e-mail. start mailto:joe.hoe@hoemail.com?subject=Notifications &; body=Hello Joe, I would like to... Start an application to write new emails, determine, subject, and body of new emails. start mailto:joe.hoe@hoemail.com?subject=Notification&body=Hello Joe,%0a%0aI'd like to... As above, with a new line included as %0a. start/b TODO:example-application-everywhere useful Starting an application without opening a new console window, redirecting output to the console from which the command starts to call. Link: TIME[edit] Displays or sets system time. Link: the time ss64.com Microsoft TITLE Sets the title displayed in the console window. Link: The ss64.com Microsoft TYPE prints the contents of the file or file to the output. Example: type a file name.txt type a.txt b.txt type *.txt NUL > tmp.txt Create a blank file (blank file). Link: type ss64.com Microsoft VER[edit] Indicates the command processor or operating system version. C:\>VER Microsoft Windows XP [Version 5.1.2600] C:\> Several versions: Microsoft Windows [Version 5.1.2600] Microsoft Windows [Version 6.0.6000] ... The word version appears local. Link: VERIFY Set or clear settings to verify the same copy the files etc. are written correctly. Link: verify on ss64.com on Microsoft VOL VOL volume labels. Link: vol on ss64.com vol in microsoft External commands[edit] External instructions available for Windows command translator are separate executable program files, supplied with operating systems by Microsoft, or combined as standard with third-party command interpreters. By replacing program files, the meaning and functionality of this command can be changed. Many, but not all, external instructions support the convention/? , cause them to write online usage information to their standard output and then go out with the status code 0. ARP[edit] Displays or changes the items in the address resolution protocol cache, which maps the IP address to a physical address. Link: arp ss64.com at ARP Microsoft AT[edit] Schedules the program to be run at a certain time. See also SCHTASKS. Link: at ss64.com Microsoft ATTRIB[edit] Displays or sets file attributes. Without arguments, it displays attributes of all files in the current directory. Without attribute modification instructions, it displays file attributes and directories that match the specifications of the given search wildcards. Just like other operating system kmods. Modification instructions: To add an attribute, attach '+' in front of the letter. To remove the attribute, attach '-' in front of its letter attribute: A - Archive H - Hidden S - System R - Read only ... and perhaps someone else. Examples: attrib Displays attributes of all files in the current directory. File attrib .txt Displays file attributes. attrib +r File.txt Add a Read-only attribute to a file. attrib -a .txt Removes archived attributes from a file. attrib -a+r File.txt Removes the Archived attribute and adds a Read-only attribute to the file. attrib +r* .txt attrib /S+ r* .txt in the subdirectory. For more information, type the attrib/?. Links: attribs at ss64.com attribs on Microsoft BCDEDIT[edit] (Not in XP). Edit boot Configuration Data (BCD) file. For more information, type bcdedit /?. Link: uncedited ss64.com Microsoft CACLS[edit] Shows or changes the discretionary access control list (DACLs). See also ICACLS. For more information, type cacls/?. Link: cacls on ss64.com Microsoft CHCP[edit] Display or set an active code page number. For more information, type chcp/?. Link: chcp ss64.com chcp at Microsoft CHKDSK[edit] Checks the disk for disk problems, lists them and repairs them if you wish. For more information, type chkdsk /?. Link: chkdsk ss64.com chkdsk at Microsoft CHKNTFS[edit] Indicates or specify whether a system inspection should be carried out when the computer is started. System checks are done using Autochk.exe. The NTFS section of the command name is misleading, because the command works not only with the NTFS file system but also with the FAT and FAT32 file systems. For more information, type chkntfs/?. Link: chkntfs on ss64.com chkntfs in Microsoft CHOICE[edit] the user selects one of the multiple options by pressing the single key, and setting the error level according to the options Absent in Windows 2000 and Windows XP, it was reintroduced in Windows Vista, and has remained on Windows 7 and 8. Example: option / m Do you agree to Present users with questions yes/ no, set the error level to 1 for yes and 2 not to. If the user presses Control + C, the error level is 0. option /c rgb/m Which color you like to present users with questions, and show letters for users. Respond to the user pressing r, g or b, set the error level to 1, 2 or 3. The alternative is set /p; see SET. Links: optional ss64.com Microsoft CIPHER[edit] Indicates the state of encryption, encryption or decrypting folders on the NTFS volume. Link: the chipher on ss64.com chipher at Microsoft CLIP[edit] (Not in XP, or make copies from Server 2003) Putting the pipe input onto the clipboard. Example: set | clip Places the environment variable listing to the clipboard. clips < File1.txt Put file1 content.txt to the clipboard. Link: clips on ss64.com microsoft CMD[edit] Using other examples of Microsoft CMD Links: cmds on ss64.com cmd at Microsoft COMP[edit] Compare files. See also FC. Link: at ss64.com Microsoft COMPACT[edit] Shows or changes the compression of files or folders on the NTFS partition. Link: CONVERT[edit] Converts the volume from fat16 or FAT32 file system to NTFS file system. Link: converting ss64.com microsoft DEBUG enables to interactively check the contents of files and memory in the installation language, hexadecimal or ASCII. Available in Windows 32-bit including Windows 7; availability in Windows 64-bit is unclear. In modern Windows, useful as a quick hack to see the contents of the file hex. Keywords: dumping hex, hexdump, hexadecimal pile, see hexadecimal, dismantling. Debug offers its own line of instructions. Once on his instructions as, type? to find about debug orders. To view a file hex, use debug.exe by file name as a parameter, and then repeatedly type d followed by entering on the debug command line. Limitations: As a DOS program, debugging is choked on a long file name. Use dir/x to find a file name 8.3, and use debug at the one. Debug could not see larger files. Link: DISKCOMP[edit] Compares the contents of two flowers. Link: the disk on ss64.com diskcomp at Microsoft DISKCOPY[edit] Copying the content of one floppy to another. Link: diskcopied ss64.com microsoft DISKPART[edit] Show and configure disk partition properties. Links: diskpart at ss64.com diskpart in Microsoft, for diskpart XP at Microsoft DOSKEY[edit] Above all, creating macros known from other operating systems as aliases. Moreover, provide functionality related to command history, and enhanced command line editing. Macro alternatives to very short group scripts. Examples related to macros: doskey da=dir /s / b Creating a single macro called da doskey np=notepad $1 Creates a single macro that passes its first argument for notepad. doskey / macrofile=doskeymacros.txt macrofile=doskeymacros.txt macro definitions from files. doskey/macro lists all macroes defined by their definition. doskey / macros | find da List all macro definitions containing da as substring; see also SEARCH. Examples related to order history: doskey/history Lists the complete order history. doskey / history | find the list of each command history row containing dir as substring/listsize doskey=100 Set the order history size to 100. For help on the doskey from the command line, type doskey/?. Link: doskey ss64.com doskey at Microsoft DRIVERQUERY[edit] Shows all installed device drivers and their properties. Link: driving in ss64.com vehicle at Microsoft EXPAND[edit] Extract files from a compressed .cab file. See also #MAKECAB. Links: ss64.com at Microsoft FC compare files, displaying their content differences in a specific way. Example: fc File1.txt File2.txt >NUL && Echo Same || different echoes or errors Detect differences using fc error level. Zero error level means that the file is the same; not zero can mean files are different but also that one of the files does not exist. Link: fc at ss64.com fc at Microsoft FIND[edit] Find strings in files or inputs, removing matching rows. Unlike FINDSTR, it can't recurringly search for folders, can't find regular expressions, require quotation marks around sought-after strings, and treat space literally rather than as logical or. Example: search (object *.txt dir /S/B | find dir/S/B receipts | search / I/V Print receipts all miss corresponding lines in the dir command output, ignoring the case of letters. If more than one file is searched, the output of one count number per file is rowed with a series of d alternates followed by a file name; did not remove the corresponding number of lines in all files. search / C/V & file.txt Output number of AKA count rows in .txt. Does wc-l work other operating systems. Serves by treating as a string that is not available on the line. Redirection usage prevents file names from output before the number of rows. file type.txt | find /C/V As above, with different syntax. type *.txt 2>NUL | find /C/V Output total file row count that ends in the .txt in the current folder. 2>NUL is a standard error removal that removes the file name followed by a blank line from the output. find Schönheit *.txt If running from a group file stored in unicode UTF-8 encoding, search for Schönheit search terms in your encoded UTF-8 file *.txt. For this to work, group files can't contain bytes order marks written by Notepad when saving in UTF-8. Notepad++ is an example of a program that allows you to write plain text files encoded UTF-8 without a bytes order mark. Despite this by searching for commands, it doesn't work #FINDSTR. seeking Copyright Working with Binary Files is not less than a text file. Links: search ss64.com Microsoft FINDSTR[edit] Find regular expressions or text strings in files. Do some known grep command jobs from other operating systems, but are far more limited in the usual expressions it supports. Treating space in common expression as a logical AKA inconsistency or unless prevented by /c option. Example: findstr/s [0-9][0-9].*[0-9][0-9] *.h *.cpp Searches repeatedly all files whose names end with dot h or dot cpp, printing only lines containing two consecutive decimal digits followed by anything followed by two digits of consecutive decimals followed by two digits of consecutive decimals followed by anything followed by two digits of consecutive decimals followed by two digits findstr a.*b a.*c Files.txt Output of all rows in File.txt which matches any two common expressions separated by space. Therefore, the effect is one of the logic or on a common expression. echo world | findstr hello wo.ld No match. Since the first item before space doesn't look like a registrar, findstr treats the entire search term as a common search term. echo world | findstr/r hello wo.ld Match. Use /r force regex treatment. findstr /r/c:ID: *[0-9]* File.txt Output of all rows in File.txt which matches a single common expression containing space. Use /c prevents space from being treated as logical or. Use /r changes the usual expression treatment, which is disabled by default by use/c. To test this, try the following: echo ID: 12|findstr /r/c:ID: *[0-9]*$echo ID: 12|findstr/c:ID: *[0-9]*$ No match, because search strings aren't interpreted as regular expressions. echo ID: abc|findstr ID: *[0-9]*$ Match even if echo output fails to match complete common expressions: searches are interpreted as one for lines matching ID: or *[0-9]*$. findstr/ric:id: *[0-9]* Files.txt Is the same as the previous example, but in an insensitive way. Although findstr allows this sort of switch accumulation behind one/, this is not possible with any order. For example, dir/bs doesn't work, while dir/b/s don't. To test this, try the following: echo ID: 12|findstr/ric:id: *[0-9]*$echo ID: ab|findstr/ric:id: *[0-9]*$findstr/msric:id: *[0-9]* *.txt Like above, but repeated for all files each/s, displays only matching files instead of the corresponding rows per /m. echo hel lo | findstr/ c:hel lo / c:world/ c switch can be used several times to create logic or. echo \hello\ | findstr \hello\ No match. Backslash before quotation marks and some other characters act as escape; therefore, \ corresponds to . echo \hello\ | search \\hello\\ Match. Double backslash passed to find a stand for a single backslash. echo \hello\ | findstr \hello\ Match. None of the single backslashes passed for findstr followed by character where the backslash acts as an escape. echo ^hey | findstr \^hey | more To find the price call (quotation mark), you need to escape twice: once to shells use caret (^), and once for findstr using backslash (\). echo ^hey | findstr ^\^hey there^ | To find a quote and have a search term included in the quote as well, the acryped quotes should be flown for shells using caret (^). online echo / / comment | findstr \/\/ If slash forward (/) is the first character in search terms, it needs to be synced with backslash (\). An escape is required even if a search term is included in the quote. findstr/f:FileList.txt def.*() Search in the file specified in File List.txt, one file per row. The file name in FileList.txt contains spaces and should not be surrounded with quotation marks for this to work. findstr/g:SearchTermsFile.txt *.txt *Find search terms available in SearchTermsFile.txt, a search term per line. Space doesn't work to separate two search terms; on the other hand, each row is a complete search term. Lines are matched if at least one of the search terms matches. If the first search term looks like a regex, the search will be a regex one, but if it looks like a regular search term, the entire search will be common even if the 2nd or newer search term looks like a regex. findstr / xlg:File1.txt File2.txt Outputs intersection set: rows found in both files. findstr / xlvg:File2.txt File1.txt Outputs sets the difference: File1.txt - File2.txt. findstr /m Microsoft C:\Windows\system32*.com works with binary files no less than text files. Common expression limitations findstr, as opposed to grep: No group support -- \(, \). No support from greedy iterators -- *?. No zero support or one of the previous -- ?. And more. Other limitations: There are various limitations and strange behaviors as documented on What are the features and limitations of windows FINDSTR commands that are not documented?. Bugs: echo bb|findstr bb baaa Found nothing in various versions of Windows, but it should be. Also consider typing findstr/?. Link: FORFILES search for files by their modification date and filename patterns, and execute commands for each file found. Very limited, especially compared to finding other operating system instructions. Available since Windows Vista. For more information, type forfiles /?. Example: forfiles / s / d 06/10/2015 / c cmd /c echo @fdate @path For each file in the current folder and its subfolders are modified on June 10, 2015 or later, output the file modification date and full file path. The date format after/d is locally specific. Therefore, enables to find the most recently modified files. Keywords: the most recent files changed. forfiles / m * .txt / s / d 06/10/2015 / c cmd / c echo @fdate @path As above, but only for files that are at .txt. Link: to fill in ss64.com at forfiles Microsoft FORMAT[edit] Disk format to use Windows-supported file systems such as FAT, FAT32 or NTFS, therephone the contents of the previous disk. To be used with caution. Link: format ss64.com format microsoft FSUTIL[edit] Powerful tools actions associated with FAT and NTFS file systems, which are used only by powerusers with extensive knowledge of the operating system. Link: GPRESULT Displays group policy settings and more for users or computers. Link: gpresult on ss64.com gpresult in Microsoft Wikipedia:GRAFTABL Group Policy[edit] Allows the display of advanced character sets in graphics mode. For more information, type the graftabl /?. Link: HELP[edit] Shows command help. Examples: help Shows a list of commands that Windows provides. A copy of the help Shows help for COPY instructions, also available by typing copy /?. Links: help ss64.com Microsoft ICACLS[edit] (Not in XP) Shows or changes the discretionary access control list (DACLs) of files or folders. See also CACLS. Fore more, type icacls/?. Link: icacls ss64.com Microsoft IPCONFIG[edit] Displays Windows IP Configuration. Shows the configuration with the connection and the name of the connection (i.e. the Local Area Connection Ethernet adapter) Below that specific information related to the connection is displayed such as the drowning address and the DNS ip address and the subnet mask. Link: ipconfig at ss64.com ipconfig at Microsoft LABEL Add, set or remove disk labels. Link: the label on ss64.com microsoft MAKECAB[edit] puts the file into a compressed .cab file. See also #EXPAND. Link: makecab at ss64.com makecab in Microsoft MODE[edit] Multipurpose command to display device status, configure ports and devices, and more. Examples: Output Status mode and configure all devices, such as com3 and con status. Output cone mode and configuration of configuration devices, the console in which the command interpreter is in progress. con cols mode=120 rows=20 Set the number of columns and lines for the current console, resulting in the resize of the window, and emptying the screen. Settings do not affect the new elements of the console. Keywords: widescreen, wide window, screen size, window size, screen resize, resize windows. mode 120, 20 As above: Sets the number of columns (120) and lines (20), resulting in the resize of the window, and emptying the screen. con cols mode=120 Set the number of columns for the current console, generates a window resize, and clears the screen. It seems to change the number of visible lines as well, but the number of console buffer counts seems unchanged. mode 120 As above: Sets the number of columns. mode con cp Output code page during console. con cp select mode=850 Set the code page during console. For a list of code pages, see the Microsoft documentation linked below. con rate mode=31 delay=1 Sets the rate and delay for recurring entry of characters while the key is held pressed, console. The lower the rate, the fewer repetitions per second. Link: a ss64.com features on Microsoft MORE[edit] Displays the contents of a file or file, one screen at a time. directed to the file, doing some conversions, also depending on the switch used. Example: More Tests.txt more grade -i search.*source of string.txt | more /p >Out.txt Take output of non-Windows grep commands that produce line breaks that consist exclusively of LF characters without CR characters, converting LF line separation to CR-LF line breaks. The new line of CR-LF is also known as DOS online breaks, Windows online separators, new DOS lines, new Windows lines, and the ending of the CR/LF line, as opposed to the LF line breaks used by some other operating systems. In some preparations, it seems like a gibberish output if the input contains a broken LF row and tab characters at the same time. In some preparations, for conversion, /p may not be necessary. Therefore, more will change the rest line even without /p. more/t4 Resources.txt >Target.txt converting tab characters into 4 spaces. In some setups, tab conversion occurs automatically, even without/t turning. If so, it is each default to 8 spaces. Change /e: Online documentation for more in Windows XP and Windows Vista doesn't mention the switch. Switch /e mentioned in more/? in Windows XP and Windows Vista. Each more/?, switch is supposed to enable the advanced features listed at the end of help more/? indicates the current row on pressing =. However, in Windows XP and Windows Vista, which seem to be enabled by default even without / e. Hypotheses: In Windows XP and Windows Vista, / e do nothing; it is present for compatibility reasons. Link: NET[edit] Provides various network services, depending on the instructions used. The variants available per instruction: net configuration account net computer continues clean file net group help net help net helpmsg net net print net sends net session start net stop time clean use net view links: OPENFILES[edit] Perform related actions to open files, especially those opened by other users Actions involving queries, displays, and disconnects. For more information, type open /?. Link: opens ss64.com on Microsoft PING[edit] Syntax: PING/? Hostname PING address Send ICMP/IP echo packets through the network to the specified address (or the first IP address that the hostname map set to go through a name search) and print all responses received. Example: ping en.wikibooks.org ping 91.198.174.192 ping Links: ping on ss64.com ping in Microsoft RECOVER[edit] Recover as much information as it can from damaged files on damaged disks. Links: recover on ss64.com on Microsoft REG Queries or modify Windows registry. The first argument is one of the following commands: inquiries, add, delete, copy, store, load, unlock, restore, compare, export, import, and flag. To more about the order, follow it with /?, like a reg/? Link: the reg on ss64.com Microsoft REPLACE[edit] Replaces the file in the destination folder with the same named file in the source folder. Link: replace on ss64.com documents at Microsoft ROBOCOPY[edit] (Not in XP) File copy Folder. See also XCOPY and COPY. Examples: robocopy /s C:\Windows\system C:\Windows-2\system *.dll Copys all files ending in .dll from one directory to another, replicating nested directory structures. Link: robocopy ss64.com the robocopy at Microsoft RUNDLL32[edit] Run the available function from DLL. The available DLL and its functionality differ among windows versions. Examples: rundll32 sysdm.cpl,EditEnvironmentVariables In some versions of Windows, opening a dialog to edit environment variables. Link: SC Controls Windows services, supports starting, stopping, queries, and more. Windows services are things like processes. Windows services are either hosted in their own processes or they are hosted in examples of svchost processes.exe, often with various services in the same example. The use of certain service processor time is available using the Process App that can be downloaded independently of Sysinternals, by going to the service property and then the Thread tab. Another instruction capable of controlling the service is NET. TASKLIST can list hosted services using /svc switch. Example: sc start wuauserv sc stop wuauserv sc inquiry wuauserv sc inquiry output information about all services. sc config SysMain start= disabled Make sure the SysMain service is disabled after starting. SysMain is a SuperFetch service, causing repeated hard activities by trying to guess which program loads into RAM if they are going to be used, and loading them. Note the mandatory deficiency of space before = and mandatory space after =. Link: sc at ss64.com Windows 7 Disc ss64.com sc microsoft SCHTASKS[edit] Schedules programs that will run at a certain time, more powerful than AT. Link: schtasks ss64.com on Microsoft SETX[edit] Like SET, but affect the entire machine instead of the current console or process. Unavailable in Windows XP; available in Windows Vista and then. Link: setx ss64.com Microsoft, Windows Server 2008, Windows Vista SHUTDOWN[edit] Shuts down the computer, or logs off the current user. Example: closure/closure/s/t 0 Closes the computer with zero delay. closure /l Logs from the current user. Link: closure on ss64.com Microsoft SORT[edit] Sorts alphabetical, from A to Z or Z to A, insensitive cases. Can not sort the proprietaries: if the input contains one integer per row, 12 comes before 9. Examples: sort .txt Remove the contents of the sorted Files.txt. compile / r .txt Reverse Order, Z to A. dir /b | sort Links: sort ss64.com microsoft SUBST[edit] Assign driver letters to local folders, display current tasks, or remove tasks. Example: subst p: . Assign p: to the current folder. subst Shows all tasks made earlier using subst /d p: substation in ss64.com subst card at Microsoft SYSTEMINFO[edit] Shows its computer configuration and operating system. Link: systeminfo ss64.com software at Microsoft TASKKILL[edit] End one or more Example: taskkill /im AcroRd32.exe Finish all processes with the name AcroRd32.exe; therefore, finish all open examples of Acrobat Reader. Names can be found using a list of tasks. taskkill /f /im AcroRd32.exe As above, but forced. Successfully complete several processes that do not end without / f. list of tasks | searching notepad taskkill /PID 5792 To complete the AKA task process with process ID (PID) 5792; if only you had encountered a PID using a list of tasks. Link: taskkill in ss64.com in Microsoft TASKLIST[edit] Lists tasks, including task name and process id (PID). Example: task list | fill in the task list | search AcroRd task list | search for /C chrome.exe Describes a task number named chrome.exe, belonging to a Google Chrome browser. task list /svc | findstr svchost Outputs Windows services are hosted in the svchost process.exe with the usual information abot process. Link: a list of tasks in the ss64.com task list in Microsoft TIMEOUT[edit] Waits a specified moment, exposing the number of times left as a time fitting, allowing users to interrupt the wait by pressing the key. Also known as sleeplessness or sleep. Available in Windows Vista and later. Example: end time / t 5 Wait for five seconds, allowing the user to cancel the wait by pressing the key. timeout /t 5/nobreak Wait for five seconds, ignoring user input other than control + C. time / t 5 / nobreak >nul As above, but without output. Workaround in Windows XP: ping -n 6 127.0.0.1 >nul Wait for five moments; number after -n is the number of times to wait plus 1. Perl-based workaround in Windows XP, which requires Perl installed: perl -e sleep 5 Links: TREE[edit] Describes the subject matter of all current directory subdirectories to any repeating stage or depth. If used with suis /F, it exposes not only subdirectories but also files. Example: principal / f Includes files in the list, in addition to the directory. subject /f/a As above, but using 7-bit ASCII characters including +, - and \ to pull the principal. Staple strokes using 8-bit ASCII characters: ——winevt | —Logs | └——TraceFormats ——winrm Principal strokes using 7-bit ASCII characters: +---winevt | +---Logs | \---TraceFormat +---a large package: WHERE[edit] Output one or more file location or file name patterns, where the file or pattern does not need to state the extension if it is listed in PATHEXT, such as .exe. The coil in the current directory and in the TRAFFIC is negligent. Are there some jobs that lead some other operating systems, but are more flexible. Available in Windows 2003, Windows Vista, Windows 7, and later; not available on Windows XP. The alternative to be used with Windows XP is in the example below. There is no in-house directive, as there are no exe dot files to match. Example: where to search Output command is searching, perhaps C:\Windows\System32\find.exe. Extension .exe not need to be determined as long as it is listed in PATHEXT, which If there are more commands in the path, output the route for both. In some cases, it can remove the following:C:\Windows\System32\find.exeC:\Program File\GnuWin32\bin\find.exe for %i (search.exe) to do @echo %~$PATH:i Location output.exe on Windows XP. Names should enter .exe, unlike the commands in which. where /r . Task* Find a file whose name matches Task* repeatedly from the current folder. Just like dir/b/s Task*. The /r switch disables the search in the folder in the PATH. where *.bat Output all .bat numbers in the current directory and in the directory available in the PATH. Therefore, outputs all .bat numbers you can run without entering their full path. where ls*.bat Above, constraints are also the beginning of your .bat name. where ls* As above, but without constraints on the extension. Find lsdisks.bat, lsmice.pl, lsmnts.py in the current directory or on the route. where *.exe *.com | More Displays countless .exe and .com contents in the path and in the current folder, including those in C:\Windows\System32. where $path:*.bat Output .bat in the path but not those in the current folder unless the current folder is in the PATH. From the path, other environmental variables that contain a list of directories can be used. where $windir:*.exe Output .exe number of files found in the folder specified in the WINDIR environment variable. where to do $path:*.bat $windir:*.exe A Confederate is available. Output all files match one of two queries. where / q * .bat &information; echo Found Blocks both standard output and errors, but sets the error level, allowing testing on it. The error level is set either way, with or without/q. Link: WMIC[edit] Starting the Windows Management Instrumentation Command Line (WMIC), or with the arguments given, approves the arguments in order to WMIC. Not in Windows XP Home. For more information, type wmic /?. Example: wmic logicaldisk gets captions, descriptions List of drives (disks) can be accessed under drive letters, whether local hard disks, CD-ROM drives, removable flash drives, network drives or drives created using #SUBST. wmicControl + C Enters wmic and then interfere with it. The side effect is that the console buffer becomes very extensive, and the screen becomes resizable hornbills with the mouse as a consequence. This is the result of a wmic setting of a large number of console columns, which you can verify using the mode cone. You can achieve the same result by typing a mode of 1500. See also #MODE. wmic datafile where the name=C:\\Windows\System32\cmd.exe gets the Output Version /output value of the cmd version.exe, which should be close to the Windows version. Link: wmic at ss64.com wmic at Microsoft XCOPY[edit] Copy files and directories in a more advanced way than value shrink in Windows Vista and then favor ROBOCOPY. Taip xcopy /? to find out more, including the unthinkable number of options. Example: xcopy C:\Windows\system Copy all files, but not files in nested folders, rather than source folders (C:\Windows\system) (C:\Windows\system) folder. xcopy /s/i/C:\Windows\system C:\Windows-2\system Copy all files and folders to any nesting depth (via /s) from the source folder (C:\Windows\system) to C:\Windows-2\system, create Windows-2\system if it does not exist (via /i). xcopy /s/i/d:09-01-2014 C:\Windows\system C:\Windows-2\system As above, but only copies of files changed on September 1, 2014 or newer. Note the use of the first convention of the month even if you are in a non-US local Windows. xcopy /L/s /i/d:09-01-2014 C:\Windows\system C:\Windows-2\system As above, but in test mode via /L (list only, output only, display only). Therefore, do not do anything real copying, just list what will be copied. xcopy /s/i/i:\Windows\system*.dll C:\Windows-2\system For one of the above examples, but only copies of files ending in the .dll, including those in the nested folder. Link: xcopy ss64.com xcopy in Microsoft External links