

[SS64](#)[CMD](#)[Syntax](#)[Links](#)

Redirection

<code>command > filename</code>	Redirect <code>command</code> output to a file
<code>command >> filename</code>	APPEND into a file
<code>command < filename</code>	Type a text file and pass the text to <code>command</code>
<code>commandA commandB</code>	Pipe the output from <code>commandA</code> into <code>commandB</code>
<code>commandA & commandB</code>	Run <code>commandA</code> and then run <code>commandB</code>
<code>commandA && commandB</code>	Run <code>commandA</code> , if it succeeds then run <code>commandB</code>
<code>commandA commandB</code>	Run <code>commandA</code> , if it fails then run <code>commandB</code>

Numeric handles:

STDIN = 0 Keyboard input
 STDOUT = 1 Text output
 STDERR = 2 Error text output
 UNDEFINED = 3-9

<code>command 2> filename</code>	Redirect any error message into a file
<code>command 2>> filename</code>	Append any error message into a file
<code>(command) 2> filename</code>	Redirect any CMD.exe error into a file
<code>command > file 2>&1</code>	Redirect errors and output to one file
<code>command > file 2<&1</code>	Redirect output and errors to one file
<code>command > fileA 2> fileB</code>	Redirect output and errors to separate files

<code>command 2>&1 >filename</code>	This will fail!
---	-----------------

Redirect to [NUL](#) (hide errors)

<code>command 2> nul</code>	Redirect error messages to NUL
<code>command >nul 2>&1</code>	Redirect error and output to NUL
<code>command >filename 2> nul</code>	Redirect output to file but suppress error
<code>(command)>filename 2> nul</code>	Redirect output to file but suppress CMD.exe errors

Note, any long filenames must be surrounded in "double quotes". A CMD error is an error raised by the command processor itself rather than the program/command.

Redirection with `>` or `2>` will overwrite any existing file.

You can also redirect to a printer with `> PRN` or `>LPT1`

To prevent the `>` and `<` characters from causing redirection, [escape](#) with a caret: `^>` or `^<`

Redirect multiple lines by bracketing a set of commands:

```
(
  Echo sample text1
  Echo sample text2
) > c:\logfile.txt
```

Unicode

The CMD Shell can redirect ASCII/ANSI (the default) or Unicode (UCS-2 le) but not UTF-8. This can be selected by launching `CMD /A` or `CMD /U`

With the default settings a UCS-2 file can be converted by redirecting it (note it's the redirection not the TYPE/MORE command that makes the encoding change)

```
TYPE unicode.txt > asciifile.txt
```

European characters like ABCàéÿ will usually convert correctly, but others like £¥¦€ will become random extended [ASCII](#) characters: œ¼Ÿ?

Pipes and CMD.exe

When a command is piped with ' | *batch_command* ' this will start (or rather CALL) a new CMD.exe instance, in effect running:

```
C:\Windows\system32\cmd.exe /C /S /D "batch_command"
```

This has several side effects:

Any newline (CR/LF) characters in the *batch_command* will be turned into & operators. (an [in depth discussion](#) of this on StackOverflow)
If the *batch_command* includes any caret escape characters ^ they will need to be doubled up so that the escape survives into the new CMD shell.

Calling a new CMD shell also has a small effect on performance, this may not be noticable for small data sets.

Examples:

```
DIR >MyFileListing.txt

DIR /o:n >"Another list of Files.txt"

DIR C:\ >List_of_C.txt 2>errorlog.txt

DIR C:\ >List_of_C.txt & DIR D:\ >List_of_D.txt

ECHO y| DEL *.txt

ECHO Some text ^<html tag^> more text

ECHO DIR C:\ ^> c:\logfile.txt >NewScript.cmd

COPY nul empty.txt

MEM /C >>MemLog.txt

Date /T >>MemLog.txt

SORT < MyTextFile.txt

SET _output=%_missing% 2>nul

FIND /i "Jones" < names.txt >logfile.txt

(TYPE logfile.txt >> newfile.txt) 2>nul
```

"Stupidity, outrage, vanity, cruelty, iniquity, bad faith, falsehood / we fail to see the whole array when it is facing in the same direction as we" ~ Jean Rostand (French Historian)

Related:

Syntax

TYPE - Display the contents of one or more text files

[Q245031](#) - Error when using the | pipe symbol

Equivalent bash command (Linux): [Redirection](#) - Spooling output to a file, piping input

Get Your Google Back

Get the Google App and Chrome and
Start Searching. Download Now

www.google.com

AdChoices 



