

[Work Port](http://www.indeed.ie/port)
www.indeed.ie/port
 Find Work Port here, the most complete job site online.



Redirection

Redirection

<i>command > file</i>	Write standard output of <i>command</i> to <i>file</i>
<i>command 1> file</i>	Write standard output of <i>command</i> to <i>file</i> (same as previous)
<i>command 2> file</i>	Write standard error of <i>command</i> to <i>file</i> (OS/2 and NT)
<i>command > file 2>&1</i>	Write both standard output <i>and</i> standard error of <i>command</i> to <i>file</i> (OS/2 and NT)
<i>command >> file</i>	Append standard output of <i>command</i> to <i>file</i>
<i>command 1>> file</i>	Append standard output of <i>command</i> to <i>file</i> (same as previous)
<i>command 2>> file</i>	Append standard error of <i>command</i> to <i>file</i> (OS/2 and NT)
<i>command >> file 2>&1</i>	Append both standard output <i>and</i> standard error of <i>command</i> to <i>file</i> (OS/2 and NT)
<i>commandA commandB</i>	Redirect standard output of <i>commandA</i> to standard input of <i>commandB</i>
<i>commandA 2>&1 commandB</i>	Redirect standard output <i>and</i> standard error of <i>commandA</i> to standard input of <i>commandB</i> (OS/2 and NT)
<i>command < file</i>	<i>command</i> gets standard input from <i>file</i>
<i>command 2>&1</i>	<i>command</i> 's standard error is redirected to standard output (OS/2 and NT)
<i>command 1>&2</i>	<i>command</i> 's standard output is redirected to standard error (OS/2 and NT)

Notes: (1) Where the table mentions redirection *to* a *file* you may also use redirection to a [device](#). Redirection *from* a device is not always possible.

(2) Redirection to the NUL device is often used to hide standard output, instead of displaying it on screen:

```
COPY *.* A: > NUL
```

Another frequently used redirection is redirection to a parallel port to print standard output:

```
DIR > LPT1
```

In COMMAND.COM, if you frequently use this kind of redirection, you may find that after some time you'll get unexpected error messages complaining that there aren't enough free file handles left to accomplish some tasks and that you should increase the number of file handles, set in CONFIG.SYS (FILES=*nn*).

Since DOS treats devices like AUX, COM*n*, LPT*n*, NUL and PRN as files, opening a device will claim one file handle.

However, unlike files, devices will never be closed until reboot.

To make things worse, each device exists in every directory on every drive, so if you used redirection to NUL in, say, C:\ and after that you use it again in C:\TEMP, you'll lose another file handle.

There are tricks to decrease the number of file handles lost by redirection:

- redirect to (one single temporary) file instead of NUL
- specify a directory if you *have to* redirect to a device:

```
DIR > %TEMP%\LPT1
```

 now you only claim one file handle per device, instead of one file handle per device per directory
- use PRINT to print *files*:

```
PRINT AUTOEXEC.BAT
```

 however, this will cost you some base memory, since a portion of PRINT stays resident in memory

These restrictions are not so severe in OS/2 or NT DOS sessions, since you'll probably close those sessions after executing only a few command(s), and a new DOS session will start with the full set of file handles available.

- (3) Redirections to one or more files tend to make batch files hard to read. Sometimes the lines can be padded with spaces to align all redirection signs and make the batch file more readable. However, if you were to do this with ECHO command lines, the spaces would really be ECHOed, which is not always convenient, to say the least.

On [Marc Stern \(http://xset.tripod.com/tip8.htm\)](http://xset.tripod.com/tip8.htm)'s web site I found a great solution: just place the redirections *before* the actual commands.

Take this imaginary batch file, for example:

```
ECHO Directory of all files on C: >> LOG1.LOG
DIR C:\ /S >> LOG1.LOG
```

Not exactly easy on the eye, that one?

How about this one, then?

```
>> LOG1.LOG    ECHO Directory of all files on C:
>> LOG1.LOG    DIR C:\ /S
```

It will do exactly the same, no difference! Much better, isn't it?

But now, try these:

```
VER | TIME > LOG1.LOG
> LOG1.LOG VER | TIME
```

As you will notice, in the second line, it is the output of VER that gets redirected to LOG1.LOG!!

As a rule of thumb: do not use this technique in command lines that also contain other redirections.

- (4) Redirecting both standard output and standard error to the same file or device is done by adding 2>&1 to the command line. This will only work in OS/2 and NT, not in MS-DOS. Where you put 2>&1 is rather critical. It will only do what it is supposed to do when placed *at the end of the command line* (as Jennie Walker pointed out to me) or right before the next pipe (|).
- (5) When using redirection to create temporary batch files, keep in mind that the output that you redirect may vary with different language versions. A sample of these differences is shown on the [DATE/TIME](#) page.
- (6) Sometimes we need redirection to create a temporary batch file that uses redirection itself. This may seem quite a challenge. How, for example, are you going to append the following command line to a temporary batch file:

```
DIR | FINDSTR /R /I /C:" 0 Dir(s)" >NUL
```

The following code will definitely **not** work:

```
ECHO DIR | FINDSTR /R /I /C:" 0 Dir(s)" >NUL >> TEMPORARY.BAT
```

For Windows NT 4 and later, you will need to [escape](#) the pipe and redirection symbols, which is done by prefixing them with carets (^):

```
ECHO DIR ^| FINDSTR /R /I /C:" 0 Dir(s)" ^>NUL >> TEMPORARY.BAT
```

What we have done is tell the ECHO command *not* to *interpret* the pipe and the first > symbol but instead to treat them as ordinary characters that must be ECHOed. The escape characters themselves will *not* be visible in the ECHOed line, so the temporary batch file will contain the normal, unescaped pipe and redirection symbol again.

For "real" DOS, (ab)use the [PROMPT](#) codes [\\$L](#), [\\$G](#) and [\\$B](#) to display and/or redirect pipe and redirection symbols.

Redirection usually results in [temporary files](#). Some notes on this subject can be found on my [Temporary Files](#) page.

Read my explanation of [standard output and standard error streams](#).

Take a look at some of the [examples](#) available, they will give you an impression of the many possibilities of redirection