

[bash](#), [howto](#), [iredirection](#), [io](#), [redirection](#), [stdin](#), [stdout](#), [stderr](#)

I/O Redirection

Through the command shell we can redirect the three standard file streams so that we can get input from either a file or another command instead of from our keyboard, and we can write output and errors to files or send them as input for subsequent commands.

For example, if we have a program called `do_something` that reads from `stdin` and writes to `stdout` and `stderr`, we can change its input source by using the less-than sign (`<`) followed by the name of the file to be consumed for input data:

```
do_something < input-file
```

If you want to send the output to a file, use the greater-than sign (`>`) as in:

```
do_something > output-file
```

Because `stderr` is not the same as `stdout`, error messages will still be seen on the terminal windows in the above example.

If you want to redirect `stderr` to a separate file, you use `stderr`'s file descriptor number (2), the greater-than sign (`>`), followed by the name of the file you want to hold everything the running command writes to `stderr`:

```
do_something 2> error-file
```

A special shorthand notation can be used to put anything written to file descriptor 2 (`stderr`) in the same place as file descriptor 1 (`stdout`): `2>&1`

```
do_something > all-output-file 2>&1
```

`bash` permits an easier syntax for the above:

```
do_something >& all-output-file
```

~~DISCUSSION~~

From:
<https://wiki.nanoscopic.de/> - nanoscopic wiki

Permanent link:
<https://wiki.nanoscopic.de/doku.php/pages/howtos/bash/bash-io-redirection>

Last update: **2021/12/09 22:00**

