

# Standard Streams

Standard streams are preconnected input and output communication channels between a computer program and its environment when it begins execution.

The three input/output connections are called `stdin` (standard input), `stdout` (standard output) and `stderr` (standard error).

When a command is executed via an interactive shell, the streams are typically connected to the text terminal on which the shell is running, but can be changed with redirection or a pipeline.

## Stdin

Standard input is a stream from which a program reads its input data. The program requests data transfers by use of the `read` operation.

Unless redirected, standard input is inherited from the parent process. In the case of an interactive shell, that is usually associated with the keyboard.

For processes that use a file as input, that file data may be passed (or piped) in through the `stdin` channel.

### Input via Stdin

Piping an input file `myVolumes.csv` via `stdin` to `TMSS10Sync`.

```
TMSS10Sync -S user:-password@serveraddress < myVolumes.csv
```

## Stdout

Standard output is a stream to which a program writes its output data.

Unless redirected, standard output is inherited from the parent process.

### Output Via Stdout

Redirecting `stdout` from `TMSS10Sync` to file `sync_stdout.txt`. If the file `sync_stdout.txt` already exists, it will be overwritten.

```
TMSS10Sync -S user:-password@serveraddress < myVolumes.csv > sync_stdout.txt
```

Redirecting `stdout` from `TMSS10Sync` to file `sync_stdout.txt`. Using the double `»` symbol instead of a single `>`, the file `sync_stdout.txt` will be appended to rather than overwritten.

```
TMSS10Sync -S user:-password@serveraddress < myVolumes.csv >>
sync_stdout.txt
```

## Stderr

Standard error is another output stream typically used by programs to output error messages or diagnostics. It is a stream independent of standard output and can be redirected separately.

### Output Via Stderr

Redirecting stderr from TMSS10Sync to file sync\_stderr.txt. If the file sync\_stderr.txt already exists, it will be overwritten.

```
TMSS10Sync -S user:-password@serveraddress < myVolumes.csv > sync_stdout.txt
2> sync_stderr.txt
```

Redirecting stdout from TMSS10Sync to file sync\_stderr.txt. Using the double » symbol instead of a single >, the file sync\_stderr.txt will be appended to rather than overwritten.

```
TMSS10Sync -S user:-password@serveraddress < myVolumes.csv >>
sync_stdout.txt 2>> sync_stderr.txt
```

## Concatenating Stdout and Stderr

Depending on the purpose of the output from stdout and stderr it is possible to combine both streams to the one output file.

```
TMSS10Sync -S user:-password@serveraddress < myVolumes.csv > 2>&1
sync_stdout_stderr.txt
```

From:

<https://rtfm.tapetrack.com/> - **TapeTrack Documentation**

Permanent link:

<https://rtfm.tapetrack.com/technote/std?rev=1626924156>

Last update: **2025/01/21 22:07**

