

File Descriptors and Redirection

When a new process is spawned, three file descriptors are also created:

<u>Name</u>	<u>Symbol</u>	<u>Default Device</u>
Standard In	0<	Keyboard
Standard Out	1>	Monitor
Standard Error	2>	Monitor

Standard Out and Standard Error can be sent (redirected) to devices other than the monitor.

Standard In can be received (redirected) from a device other than the keyboard.

Redirecting Standard Out and Standard Error

Assume that File1 exists, and File2 does not.

Display the contents of File1

```
$ cat File1
This is File1
$
```

Display the contents of File2 then File1

```
$ cat File2 File1
cat: File2: No such file or directory
This is File1
```

What Happened?

cat sent an error message through Standard Error to the monitor then output the contents of File1

Now we'll attempt to redirect the contents of File1 and File2 to the file holder.txt

```
$ cat File2 File1 > holder.txt  
cat: File2: No such file or directory
```

What Happened?

cat sent an error message through Standard Error to the monitor then redirected the contents of File1 to holder.txt

Display contents of holder.txt

```
$ cat holder.txt  
This is File1
```

When Standard Out is sent through a pipe, Standard Error does NOT go through the pipe

```
$ cat File2 File1 | tr "[a-z]" "[A-Z]"  
cat: File2: No such file or directory  
THIS IS FILE1
```

What Happened?

cat sent an error message through Standard Error to the monitor then sent the contents of File1 through a pipe to the translate filter who's output was sent to the monitor.

Send Standard Out and Standard Error to different files

```
$ cat File2 File1 > holder.txt 2> error.txt
```

What Happened?

cat produced an error message that was redirected to the file error.txt. The contents of File1 were redirected to the file holder.txt.

```
$ cat holder.txt  
This is File1
```

```
$ cat error.txt  
cat: File2: No such file or directory
```

Send Standard Out and Standard Error to the same file

NOTE: The construct `2>&1` means redirect Standard Error to the same place as Standard out...or, Standard Error becomes a duplicate of Standard Out

```
$ cat File2 File1 > holder.txt 2>&1
```

```
$ cat holder.txt  
cat: File2: No such file or directory  
This is File1
```

NOTE: Order is important.

```
$ cat File2 File1 2>&1 > holder.txt
```

Problem:

**Standard Error is redirected to Standard Out
BEFORE Standard Out is redirected to holder.txt**

**That means that Standard Out is still defaulting to
the monitor at the time Standard Error is redirected
to Standard Out**

**Standard Error goes to the monitor. Standard Out
goes to the file holder.txt**

Now we can send both Standard Out and Standard Error through a pipe

```
$ cat File2 File1 2>&1 | tr "[a-z]" "[A-Z]"  
CAT: FILE2: NO SUCH FILE OR DIRECTORY  
THIS IS FILE1
```

What Happened?

Since Standard Error is made a duplicate of Standard Out, everything goes through the pipe and gets translated.