

[x Dismiss](#)

Join the Stack Overflow Community

Stack Overflow is a community of 7.3 million programmers, just like you, helping each other.
Join them; it only takes a minute:

[Sign up](#)

Exit Shell Script Based on Process Exit Code



I have a shell script that executes a number of commands. How do I make the shell script exit if any of the commands exit with a non-zero exit code?

bash shell

edited Jul 31 '11 at 19:12



Cerin

18.8k 44 161 311

asked Sep 18 '08 at 6:03



Mark Roddy

12.2k 10 55 62

- 2 I answered assuming you're using bash, but if it's a very different shell can you specify in your post?
– Martin W Sep 18 '08 at 6:11

9 Answers

After each command, the exit code can be found in the `$?` variable so you would have something like:

```
ls -al file.ext
rc=$?; if [[ $rc != 0 ]]; then exit $rc; fi
```

You need to be careful of piped commands since the `$?` only gives you the return code of the last element in the pipe so, in the code:

```
ls -al file.ext | sed 's/^/xx: /'
```

will not return an error code if the file doesn't exist (since the `sed` part of the pipeline actually works, returning 0).

The `bash` shell actually provides an array which can assist in that case, that being `PIPESTATUS`. This array has one element for each of the pipeline components, that you can access individually like `${PIPESTATUS[0]}`:

```
pax> false | true ; echo ${PIPESTATUS[0]}
1
```

Note that this is getting you the result of the `false` command, not the entire pipeline. You can also get the entire list to process as you see fit:

```
pax> false | true | false; echo ${PIPESTATUS[*]}
1 0 1
```

If you wanted to get the largest error code from a pipeline, you could use something like:

```
true | true | false | true | false
rcs=${PIPESTATUS[*]}; rc=0; for i in $rcs; do rc=$((($i > $rc ? $i : $rc)); done
```