Exit Codes

What each tacctl exit status means on shift and in scripts · Checked against tacctl 0.2.25

Contents

1. The codes 2
2. In scripts 3
3. In the audit trail 3

Every tacctl command ends with an exit status. On shift you rarely need it, because the message on screen says what happened. In scripts and monitoring checks, the status is how you tell success, refusal and "a change is needed" apart.

1. The codes

Code	Meaning	What to do
0	Done, or nothing to do. Answering *no* to a removal's confirmation also cancels with nothing changed and exits 0.	Nothing
1	Refused or failed. The reason is on standard error. Nothing changed unless the message says so. Includes a check that found a problem (`config validate`, `device check`, `host check`), a device that could not be read or planned, and a Junos apply that failed, rolled back or was left pending. A command or sub-command that does not exist ends with 1 as well: on standard error one error line that names it, a `Did you mean` line when something fits and a pointer (`Run ... -h` for a sub-command), and nothing on standard output. No word, `help` and a word that starts with `-` still print the usage of a command that has them.	Read the message. For an apply, read each device's row.
2	Wrong arguments. Also, with `--exit-code`: `device config diff` found a difference, or `device config plan` / `apply --check` found a change is needed.	Fix the command, or act on the device that needs a change
3	A backend command was refused because a rendered file holds a hand edit it would discard. Nothing was written.	`tacctl config validate` names both ways out (keep or discard the edit). Usually a superuser task.
126	sudo could not be run to re-run tacctl as root	Check sudo on the server; escalate to the server owner
127	sudo is not installed, and tacctl was not run as root	As 126
130	Interrupted (Ctrl-C at a prompt, or during a pull, an apply, or `host sync` or `host check` with `--concurrency`)	For an apply, check each device's state; a pull never writes to a device

`tacctl ssh` ends with the status of the ssh it runs, and `tacctl shell -c` with the status of its line.

Two cases end with 1 although the change stands. `tacctl breakglass hold` or `release`, and `tacctl config breakglass-firewall`, exit 1 when `nft` is missing or fails to load the firewall table: the hold is recorded and rendered, and the report says in red what the server still does (*Break Glass* §2.1). Where the table would refuse nothing (the setting `off`, or nothing held), a missing `nft` is no failure and the

status is 0. `emergency lockdown enable` and `disable` exit 0 when hosts are left on the pending list, because the lockdown on the devices is in force (*Emergency Lockdown* §3.1).

---

2. In scripts

`--exit-code` turns "something differs" into status 2, which makes drift easy to check from a scheduler or a monitoring system:

```
tacctl device config diff sw1 --exit-code      # 0 = matches, 2 = differs, 1 = could not compare
tacctl device config plan sw1 --exit-code      # 2 = a change is needed
```

> A check that exits 1 (`config validate`, `device check`, `host check`) found a problem **or** could not run. Read standard error before you act.

---

3. In the audit trail

Every command that changes state records its status as `status=` on its `cmd end` line, with a word in `result=`:

`result=`	Status
`ok`	0
`interrupted`	130
`denied`	1 (the caller's tier or scope did not allow it)
`aborted`	2 (tacctl itself stopped unexpectedly)
`failed`	Any other status

Audit and Compliance §2 explains the audit lines.

See also: Troubleshooting §8 · Exit status and environment (repo) · `man tacctl` (EXIT STATUS)