

Exit Codes



- [Standard exit codes](#)
 - [Exit codes for boolean-result commands](#)
 - [Using exit codes in scripts](#)
-

WP-CLI follows widely-adopted Unix conventions for exit codes: `0` signals success (the command's primary condition was satisfied), while non-zero values signal that the primary condition was not satisfied. This can represent either an operational error (the command could not perform the requested operation) or an expected negative result that calling scripts are intended to test for (for example, "not installed"). Scripts that call WP-CLI commands can inspect the exit code to determine whether to continue, retry, or abort.

Standard exit codes

Exit code	Meaning
<code>0</code>	Command completed successfully. The requested operation was performed as expected.
<code>1</code>	Command failed. WP-CLI could not perform the operation as expected (e.g. invalid arguments, WordPress not found, a fatal error was encountered).

`WP_CLI::error()` ([doc](#)) is the conventional way to report a failure and exit with code `1`.

`WP_CLI::halt()` ([doc](#)) can be used to exit with a specific non-zero code when the command needs to communicate a particular outcome to the calling script.

Exit codes for boolean-result commands

Some commands are designed to communicate a binary result directly through the exit code rather than through output. These commands still use the standard `0/1` codes, but `1` means "condition not met" rather than "operational error". For example:

- `wp plugin is-installed <plugin>` exits with `0` if the plugin is installed, `1` if it is not.
- `wp theme is-installed <theme>` follows the same pattern.
- `wp core is-installed` exits with `0` when WordPress is installed, `1` when it is not.

Use exit code `1` to signal "condition not met" (rather than an error) when:

1. The command result is binary (pass/fail, found/not-found) and is intended to be tested directly in a shell conditional.
2. The intent is explicitly documented so script authors know the exit code carries a result meaning, not an error meaning.

Do **not** return a non-zero exit code for conditions that are informational rather than a definitive result (e.g. a list command that finds no items is still a successful operation — it returned an empty list).

Using exit codes in scripts

Because WP-CLI uses standard exit codes, you can use WP-CLI commands directly in shell conditionals:

```
# Check if a plugin is installed before trying to activate it
if wp plugin is-installed my-plugin; then
    wp plugin activate my-plugin
else
    echo "Plugin not installed"
    exit 1
fi
```

You can also inspect `$?` after any WP-CLI command to check whether it succeeded:

```
wp core update
if [ "$?" -ne 0 ]; then
    echo "Update failed"
fi
```

📄 #4

🔄 cocobook-sync 📄 2026-10-28 11:01:29 UTC

🔄 cocobook-sync 📄 2026-10-29 01:42:31 UTC