

13.2 Shell Script Conventions

[Go To Top](#)

[Return to Build and Run Scripts](#)

Shell scripts should follow predictable conventions so that developers and reviewers can understand what each script does, what it changes, and how it fails.

Each shell script should:

1. Include a file header that identifies the script purpose, expected working directory, required inputs, modified outputs, exit-code behavior, and important dependencies.
2. Validate required tools, files, directories, arguments, and environment variables before performing destructive or irreversible actions.
3. Use clear function names for repeated logic.
4. Write diagnostic output that supports troubleshooting and CI logs.
5. Return zero on success and non-zero on failure.
6. Avoid hard-coded absolute paths.
7. Avoid relying on undocumented shell aliases, shell profiles, editor settings, or user-specific environment variables.
8. Use Repository-relative paths where practical.
9. Treat missing required inputs as failures.
10. Treat missing optional inputs as warnings where appropriate.
11. Avoid destructive actions unless the script name, help text, and documentation make those actions clear.

Shell scripts should use portable shell behavior where practical. If a script requires Bash-specific features, the file header and Script Reference should say so. If the team supports Windows through WSL, PowerShell, or another shell environment, the relevant checklist instance should identify the expected shell.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:
<https://wiki.didosolutions.com/fxdemo/05-part/13-build-and-run-scripts/13-2-shell-script-conventions/start>

Last update: **2026/08/10 14:59**

