

16.3 Script Failures

[Go To Top](#)

[Return to Exception Handling](#)

Scripts should fail clearly when required conditions are not met. A script should not continue after a missing prerequisite, invalid argument, missing input file, invalid Configuration, failed generation step, failed build step, failed runtime startup, or failed validation step unless the script explicitly supports partial completion.

Script failure handling should include:

1. Clear error messages.
2. Meaningful non-zero exit codes.
3. Identification of the failed check or operation.
4. Identification of the expected input, tool, path, or condition.
5. Guidance on what the user should inspect next.
6. Preservation of useful logs or intermediate Evidence where practical.
7. Avoidance of destructive cleanup unless explicitly requested.

Scripts should distinguish warnings from failures. A missing optional tool may produce a warning. A missing required tool should cause failure. A failed required build or validation step should cause failure. A script should not return success when a required action fails.

Script failure behavior should align with the script conventions defined in Section 13. Scripts that support CI or acceptance Evidence should produce output that automated tools and reviewers can inspect.

© 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/16-exception-handling/16-3-script-failures/start>

Last update: **2026/08/10 15:44**

