

16.1 Purpose of Exception Handling Rules

[Go To Top](#)

[Return to Exception Handling](#)

Exception handling rules help the team respond to errors consistently across [Nodes](#), scripts, containers, generated [Artifacts](#), [Configuration](#), [DDS](#) communication, and validation workflows. A developer, tester, reviewer, or demonstration operator should be able to understand what failed, where it failed, why it mattered, and what the system did next.

Exception handling should support the following goals:

1. Detect errors close to their source.
2. Report errors with enough context to support diagnosis.
3. Avoid exposing secrets or sensitive local values.
4. Distinguish recoverable errors from non-recoverable errors.
5. Preserve logs and [Evidence](#).
6. Publish appropriate [Control Plane](#) status where possible.
7. Return meaningful exit codes where a process or script exits.
8. Avoid silently swallowing exceptions.
9. Avoid uncontrolled retry loops.
10. Keep failure behavior consistent with the Node [Lifecycle](#) model.

Exception handling should not make Phase 0 appear more complete than it is. If a Node, script, generated Artifact, DDS participant, container, or validation step fails, the implementation should clearly report the failure.

© 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/16-1-purpose-of-exception-handling-rules/start>

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

