

10.9 Node Error Handling

[Go To Top](#)

[Return to Node Implementation Pattern](#)

A [Node](#) should handle errors in a predictable and observable way. Error handling should distinguish Configuration errors, [DDS](#) communication errors, type-generation or binding errors, runtime dependency errors, command-handling errors, data-processing errors, and shutdown errors.

A Node should:

1. Log errors with enough context to support diagnosis.
2. Avoid exposing secrets or sensitive local values.
3. Publish Degraded, Recovering, or Failed status when the error affects Node operation.
4. Reject invalid commands or messages clearly.
5. Use meaningful exit codes when the process exits.
6. Preserve [Evidence](#) needed for testing and review.
7. Avoid swallowing exceptions without logging or status impact.
8. Avoid uncontrolled retry loops.

Recoverable errors may allow the Node to continue in the Degraded or Recovering state. Non-recoverable errors should move the Node toward a failed or controlled shutdown. The Node profile should define any Node-specific recovery behavior.

Exception handling code shall follow the file header and inline documentation conventions defined in Section 7.10. Functions or methods that raise, catch, translate, log, suppress, or propagate errors shall document the relevant error conditions and expected recovery or failure behavior.

© 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/10-node-implementation-pattern/10-9-node-error-handling/start>

Last update: **2026/08/10 09:09**

