

16.7 Recoverable and Non-Recoverable Errors

[Go To Top](#)

[Return to Exception Handling](#)

The team should distinguish recoverable errors from non-recoverable errors. This distinction helps Nodes choose between continuing, entering Degraded state, entering Recovering state, stopping, or failing.

A recoverable error allows a Node, script, or containerized process to continue safely after reporting the issue. Examples may include:

1. Unsupported command received and rejected.
2. Optional tool missing from the local environment.
3. Temporary failure to receive a non-critical message.
4. Validation warning that does not block the selected workflow.
5. Non-critical log export issue where primary logs remain available.

A non-recoverable error prevents the required operation from continuing safely. Examples may include:

1. Missing required [Configuration](#).
2. Invalid required Configuration.
3. [DDS](#) participant creation failure.
4. Required generated type missing.
5. Required container image build failure.
6. Required [Node](#) startup failure.
7. Required [Control Plane](#) status publication failure.
8. Required validation failure.
9. Unhandled exception in required runtime behavior.

The Node profile, script documentation, or validation procedure should define which errors count as recoverable for that Artifact. Developers should not make local assumptions about recoverability when the error affects shared [Baseline](#) behavior.

Recoverable errors should still produce logs, warnings, status updates, or [Evidence](#). Non-recoverable errors should fail clearly and should not leave the Baseline appearing successful.

© 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/16-7-recoverable-and-non-recoverable-errors/start>

Last update: **2026/08/10 15:47**

