

NOD-004g — Handle a Node Lifecycle Exception

[Go to NOD-004 — Control the Node Lifecycle](#)

Statement

The [DIDO-TE](#) SHALL handle each exception arising during the lifecycle of a [Node](#).

Derived From

- [NOD-004 — Control the Node Lifecycle](#)
- [NOD-004a — Establish the Initial Node State](#)
- [NOD-004b — Start the Node](#)
- [NOD-004c — Confirm Node Readiness](#)
- [NOD-004d — Transition the Node State](#)
- [NOD-004e — Stop the Node](#)
- [NOD-004f — Restart or Recover the Node](#)
- [\[DTE5\] DIDO-TE Requirements Register](#), source requirement identifier and obligation to be assigned
- [\[DTE6\] Structured Information Processing Reference Architecture \(SIP-RA\)](#)
- [\[DTE7\] Federated Data Interpretation Systems Reference Architecture \(FDIS-RA\)](#)

Rationale

A Node lifecycle exception occurs when a lifecycle operation or observed Node state differs from its authorized procedure, preconditions, expected sequence, time constraints, or intended result.

Lifecycle exceptions include an unknown initial state, failed startup, unsuccessful readiness determination, invalid state transition, interrupted stop, failed restart or recovery, unexpected state change, unavailable Dependency, lost [Node Binding](#), and disagreement between the reported and observed Node states.

Exception handling detects, identifies, assesses, contains, records, and dispositions the exceptional condition. The applicable disposition may retry an operation, return the Node to a known state, isolate the Node, initiate recovery, stop dependent activity, invalidate affected [Test Results](#), or escalate the condition to an authorized decision-maker.

Uncontrolled lifecycle exceptions may allow a failed, degraded, incorrectly configured, or otherwise unsuitable Node to remain active within the [Test Environment](#).

Applies To

- [DIDO-TE](#)
- [Node](#)
- [Node Definition](#)
- [Node Implementation](#)
- [Node Configuration](#)
- [Node Binding](#)
- [Dependency](#)
- [Test Definition](#)
- [Test Environment](#)
- [Test Resource](#)
- [Test Execution](#)
- [Test Result](#)
- [Evidence](#)
- [Traceability](#)

Verification

Verification confirms that:

- The DIDO-TE detects a lifecycle operation or Node state that differs from its authorized criteria, procedure, sequence, time constraints, or expected result.
- The DIDO-TE uniquely identifies each Node lifecycle exception.
- The exception record identifies the affected Node and lifecycle operation.
- The DIDO-TE identifies the actual Node state when the exception is detected.
- The DIDO-TE identifies the expected Node state or lifecycle outcome.
- The DIDO-TE classifies the exception according to identified exception criteria.
- The DIDO-TE determines the severity, scope, and potential effect of the exception.
- The DIDO-TE identifies affected Nodes, Dependencies, Node Bindings, Test Resources, Test Executions, Test Inputs, Test Results, and Evidence.
- The DIDO-TE prevents an affected Node from continuing lifecycle activity when continued activity presents an unacceptable risk.
- The DIDO-TE isolates the affected Node when required by the applicable exception-handling procedure.
- The DIDO-TE prevents an affected Node from participating in Test Execution unless an authorized disposition permits continued participation.
- The DIDO-TE uses an identified and authorized exception-handling procedure.
- The exception disposition identifies the responsible authority, rationale, required actions, and conditions for resolution.
- The DIDO-TE retries a lifecycle operation only when an authorized procedure permits the retry.
- The DIDO-TE prevents an unlimited or uncontrolled sequence of retries.
- The DIDO-TE places the Node in a known state or identifies the actual resulting state.
- The DIDO-TE initiates restart, recovery, stopping, isolation, rollback, or escalation when required by the authorized disposition.
- The DIDO-TE determines the validity and required disposition of each affected Test Execution and Test Result.
- The DIDO-TE records any approved exception, including its authority, rationale, scope, duration,

conditions, and expected effect.

- The DIDO-TE verifies completion of each corrective, containment, recovery, or escalation action.
- The DIDO-TE assigns a distinguishable status to each open, contained, resolved, accepted, or escalated exception.
- The DIDO-TE prevents closure of an exception without an identified disposition and supporting Evidence.
- The DIDO-TE records the exception identity, affected Node, lifecycle operation, detection time, expected state, actual state, classification, severity, scope, findings, disposition, authority, actions, resulting state, and closure status.
- The DIDO-TE maintains Traceability among the exception, Node Definition, Node Implementation, Node Configuration, Node Bindings, lifecycle procedure, affected Test Environment, Test Execution, Test Results, disposition, corrective actions, and resulting Evidence.
- An undetected, unhandled, unauthorized, unresolved, incorrectly dispositioned, or untraceable Node lifecycle exception constitutes nonconformance with this requirement.

Verification includes:

- Inspection of lifecycle exception criteria and handling procedures
- Introduction of an unsuccessful startup condition
- Introduction of an invalid lifecycle transition
- Introduction of an unexpected Node state change
- Introduction of an unavailable Dependency or Node Binding
- Confirmation that the DIDO-TE identifies and classifies each exception
- Confirmation that the DIDO-TE contains or isolates the affected Node when required
- Confirmation that uncontrolled retries do not occur
- Confirmation that an unsuitable Node cannot continue participating in Test Execution
- Inspection of exception dispositions and responsible authorities
- Confirmation that affected Test Executions and Test Results receive an identified disposition
- Inspection of corrective actions, resulting Node states, closure records, Evidence, and Traceability

Referenced By

- [NOD-004h — Record Node Lifecycle Activity](#)

Related Architecture Sections

- Add links to the architecture sections governing Node lifecycle exceptions, exception handling, Node isolation, recovery, Test Execution disposition, Evidence, and Traceability.

Delivery Phase

- ☐ Assign the applicable delivery phase.

Requirement Status

Draft

Statement Reference

Use the following syntax to reference this requirement's Statement section from another DokuWiki page:

```
{{section>dido:03-dido-te:99-annexes:annex-c-requirements:03-node-requirements:nod-004-control-the-node-lifecycle:nod-004g-handle-a-node-lifecycle-exception#Statement&noheader&nofooter&noeditbtn}}
```

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

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

Permanent link:
<https://wiki.didosolutions.com/dido/03-dido-te:99-annexes:annex-c-requirements:03-functional-requirements:03-02-node-requirements:nod-004-control-the-node-lifecycle/nod-004g-handle-a-node-lifecycle-exception?rev=1787001645>

Last update: 2026/08/17 14:20

