

C.3.3 Twin Node Requirements

[Go to C.3 Functional Requirements](#)

This section identifies the requirements governing [Twin Nodes](#) used to represent, observe, simulate, reproduce, compare, or interact with corresponding [Nodes](#) within distributed verification and testing.

Twin Node requirements address the controlled relationship between a Twin Node and its corresponding Node, including identity, association, [Configuration](#), state acquisition, synchronization, simulation, comparison, selection, controlled interaction, and [Traceability](#).

A Twin Node may receive information from multiple eligible sources. Configuration, communication policies, and applicable Quality of Service controls govern source eligibility and communication behavior without assigning semantic authority to a source solely because the source participates in the communication mechanism.

The requirements in this section derive from the applicable source documents identified in Annex B. The DIDO-TE BAA explicitly requires management of Twin Nodes and selection during testing between a virtual representation and its corresponding real-world Node. The DIDO Reference Architecture provides the broader Digital Twin, DDS publish-subscribe, and DDS Quality of Service concepts used to normalize and decompose that source requirement into individually verifiable architectural obligations.

A Twin Node requirement may be represented by an individual leaf requirement or by a non-leaf requirement group containing subordinate independently verifiable obligations.

Requirements

Requirement ID	Requirement Title	Primary Source Basis
TWIN-001	Define a Twin Node	[DTE2] , F17; [DTE9] , Appendix A, <i>Digital Twin</i>
TWIN-002	Associate a Twin Node with a Corresponding Node	[DTE2] , F17; [DTE9] , Appendix A, <i>Digital Twin</i>
TWIN-003	Configure a Twin Node	[DTE2] , F17; [DTE9] , Digital Twin and DDS Quality of Service concepts
TWIN-004	Acquire Twin Node State	[DTE9] , Digital Twin and DDS publish-subscribe concepts
TWIN-005	Synchronize Twin Node State	[DTE9] , Digital Twin and DDS Quality of Service concepts
TWIN-006	Simulate Twin Node Behavior	[DTE9] , Appendix A, <i>Digital Twin</i>
TWIN-007	Compare Twin and Corresponding Node State and Behavior	[DTE9] , Appendix A, <i>Digital Twin</i>
TWIN-008	Select a Twin or Corresponding Node for Test Execution	[DTE2] , F17
TWIN-009	Control Twin Node Interaction with the Corresponding Node	[DTE9] , Appendix A, <i>Digital Twin</i>
TWIN-010	Maintain Twin Node Traceability	[DTE2] , F17; DIDO-TE Traceability requirements

Contents

- [TWIN-001 — Define a Twin Node](#)
 - [TWIN-002 — Associate a Twin Node with a Corresponding Node](#)
 - [TWIN-003 — Configure a Twin Node](#)
 - [TWIN-004 — Acquire Twin Node State](#)
 - [TWIN-005 — Synchronize Twin Node State](#)
 - [TWIN-006 — Simulate Twin Node Behavior](#)
 - [TWIN-007 — Compare Twin and Corresponding Node State and Behavior](#)
 - [TWIN-008 — Select a Twin or Corresponding Node for Test Execution](#)
 - [TWIN-009 — Control Twin Node Interaction with the Corresponding Node](#)
 - [TWIN-010 — Maintain Twin Node Traceability](#)
-
- [TWIN-001 — Establish Twin Identity](#)
 - [TWIN-002 — Associate Twin Realizations](#)
 - [TWIN-003 — Configure a Twin Relationship](#)
 - [TWIN-004 — Select a Twin Realization](#)
 - [TWIN-005 — Access Twin Realization State](#)
 - [TWIN-006 — Synchronize Twin Realization State](#)
 - [TWIN-007 — Simulate Twin Behavior](#)
 - [TWIN-008 — Compare Twin Results](#)
 - [TWIN-010 — Control Twin Interaction](#)

Notes for Editors

Twin Node requirements use the TWIN requirement prefix.

TWIN-001, TWIN-003, TWIN-004, TWIN-005, TWIN-006, TWIN-007, TWIN-008, and TWIN-009 are initially modeled as non-leaf requirement groups because their subject areas are expected to contain multiple independently verifiable obligations.

TWIN-002 and TWIN-010 are initially modeled as leaf requirements. Review their normative Statements for atomicity before finalizing that classification.

The explicit links under Contents remain while the Twin Node requirement pages are being created and reviewed. After all planned requirement pages exist, the explicit links may be removed because the indexmenu automatically discovers them.

Apply the requirement identification, decomposition, source and derivation, normative-language, quality, status, and page-structure rules defined by Annex C.

Preserve stable requirement identifiers when moving or reorganizing Twin Node requirement pages.

Do not allocate Twin Node requirements directly to a particular DDS implementation, product, or realization within the requirement Statement. Maintain realization relationships through Traceability.

© 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/03-03-twin-node-requirements/start?rev=1787190887>
Last update: **2026/08/19 18:54**

