

5. DIDO-TE Information Model

[Go to DIDO-TE Cover Page](#)

The DIDO-TE Information Model defines the principal information elements, relationships, and structures used to describe, configure, execute, observe, and evaluate tests within the [DIDO-TE](#).

The Information Model provides the architectural vocabulary and structural relationships used by the DIDO-TE [requirements](#) and [Concept of Operations \(ConOps\)](#). It establishes what information exists and how that information is related without prescribing a particular storage technology, serialization format, programming language, database schema, or implementation platform.

The [model](#) builds upon the existing DIDO Reference Data Model and extends that model where required by the DIDO-TE architecture.

Purpose

The DIDO-TE Information Model provides a common structural basis for:

- defining the subject and scope of a test;
- organizing test plans, [Test Sets](#), [Test Cases](#), and [Test Steps](#);
- identifying [Test Executables](#) and their [Test Arguments](#);
- specifying [Expected Results](#);
- identifying the [Test Environment](#) in which testing occurs;
- representing the participants and [Resources](#) required by a test;
- specifying state, time, network, communication, [Configuration](#), and other governed test conditions;
- representing [Test Results](#) and [Test Runs](#);
- supporting [Traceability](#) among [Test Definitions](#), [Test Executions](#), Test Results, and supporting [Evidence](#); and
- providing a structural foundation for the DIDO-TE Concept of Operations and normative requirements.

Modeling Approach

The Information Model distinguishes architectural concepts from their implementation.

[Unified Modeling Language \(UML\)](#) class diagrams in this section describe information elements and their relationships. The diagrams are not intended to prescribe object-oriented software classes or a particular implementation technology.

Each model element is described using:

- its architectural purpose;
- its defining characteristics;
- its attributes, where applicable;
- its associations with other model elements;

- association multiplicities where established by the source model;
- relevant constraints and architectural interpretation; and
- Traceability to the source model and DIDO-TE requirements.

Where the existing DIDO Reference Data Model provides an element or relationship, this section preserves that source relationship unless DIDO-TE introduces a deliberate refinement.

Where DIDO-TE introduces a concept not represented in the existing model, the extension is identified explicitly rather than being represented as though it existed in the original source model.

Relationship to the DIDO Reference Data Model

The initial DIDO-TE Information Model is derived in part from the Testing Package of the DIDO Reference Data Model.

The existing model includes the following principal testing elements:

- **Test**
- **Test Plan**
- [Test Set](#)
- [Test Case](#)
- [Test Step](#)
- [Test Executable](#)
- [Expected Results](#)
- [Test Result](#)
- [Test Run](#)
- **Test Tag**

The existing model also establishes relationships between the testing structure and its execution environment.

DIDO-TE extends this foundation with architectural concepts required for governed distributed testing, including Test Environment Configuration, [Nodes](#), [Twin Nodes](#), [Virtual Networks](#), communication conditions, state control, time control, [Validation](#), Evidence, and other concepts defined by the DIDO-TE requirements.

Relationship to Test Definition

The [Test Definition](#) provides the governed specification of what is required for a test.

The existing DIDO Reference Data Model does not define a model element named Test Definition. The DIDO-TE Information Model therefore derives the structure of the Test Definition from:

- the existing testing elements and relationships in the DIDO Reference Data Model;
- the information required by the DIDO-TE functional requirements; and
- the relationships required to connect test specification, Test Execution, Test Results, Validation, and Evidence.

The Test Definition model is developed after the constituent testing elements are described so that its structure is derived from those elements rather than imposed independently of them.

Relationship to the Concept of Operations

The Information Model defines the information elements and relationships used by DIDO-TE.

The [Concept of Operations \(ConOps\)](#) describes how actors, capabilities, systems, and processes create, configure, use, exchange, and evaluate those information elements during testing.

Accordingly:

- the Information Model defines **what information exists and how it is related**; and
- the Concept of Operations describes **how that information participates in DIDO-TE operations**.

Relationship to Requirements

The DIDO-TE [requirements](#) place normative obligations on the concepts and relationships defined by the Information Model.

In particular, [C.3.6 Test Definition Requirements](#) identifies the information that a governing Test Definition is required to specify, identify, or reference.

The Information Model provides the structural basis for those requirements. Annex C remains the normative source of DIDO-TE conformance obligations.

Contents

- [5.1 Information Model Overview](#)
- [5.2 Test](#)
- [5.3 Test Plan](#)
- [5.4 Test Set](#)
- [5.5 Test Case](#)
- [5.6 Test Step](#)
- [5.7 Test Executable](#)
- [5.8 Expected Results](#)
- [5.9 Test Result](#)
- [5.10 Test Run](#)
- [5.11 Test Tag](#)
- [5.12 Test Environment Relationships](#)
- [5.13 Test Definition](#)
- [5.14 Integrated DIDO-TE Information Model](#)

Source

The initial structure of this section is derived from the Testing Package of the DIDO Reference Data Model and is refined by the DIDO-TE architecture and requirements.

Notes for Editors

- Treat the DIDO Reference Data Model as the source for existing modeled elements and relationships.
- Do not silently correct anomalies in the source model. Identify corrections or architectural refinements explicitly.
- Use focused UML class diagrams for individual model elements before producing the integrated Information Model.
- Preserve source-model multiplicities where they are known.
- Distinguish source-model content from DIDO-TE extensions.
- Do not treat UML classes as implementation classes.
- Develop the Test Definition structure from the constituent model elements and the obligations identified by the DIDO-TE requirements.
- Use the integrated diagram in Section 5.14 to show the resulting DIDO-TE Information Model after the individual elements have been reviewed.

© 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/06-information-model/start?rev=1787530015>

Last update: **2026/08/23 17:06**

