

6.2 Test

[Go to 6. DIDO-TE Information Model](#)

A **Test** is the top-level testing element represented by the Testing Package of [\[DTE4\] DIDO Reference Data Model](#).

Within [DTE4], a Test provides identifying and descriptive information, may contain one or more **Test Plans**, may be associated with [Test Results](#), and may be associated with classification tags.

The DIDO-TE Information Model preserves the [DTE4] Test element while its relationship to the [Test Definition](#) is evaluated.

Class Diagram

Figure 6.2-1 presents the Test element and its immediate relationships in [DTE4].

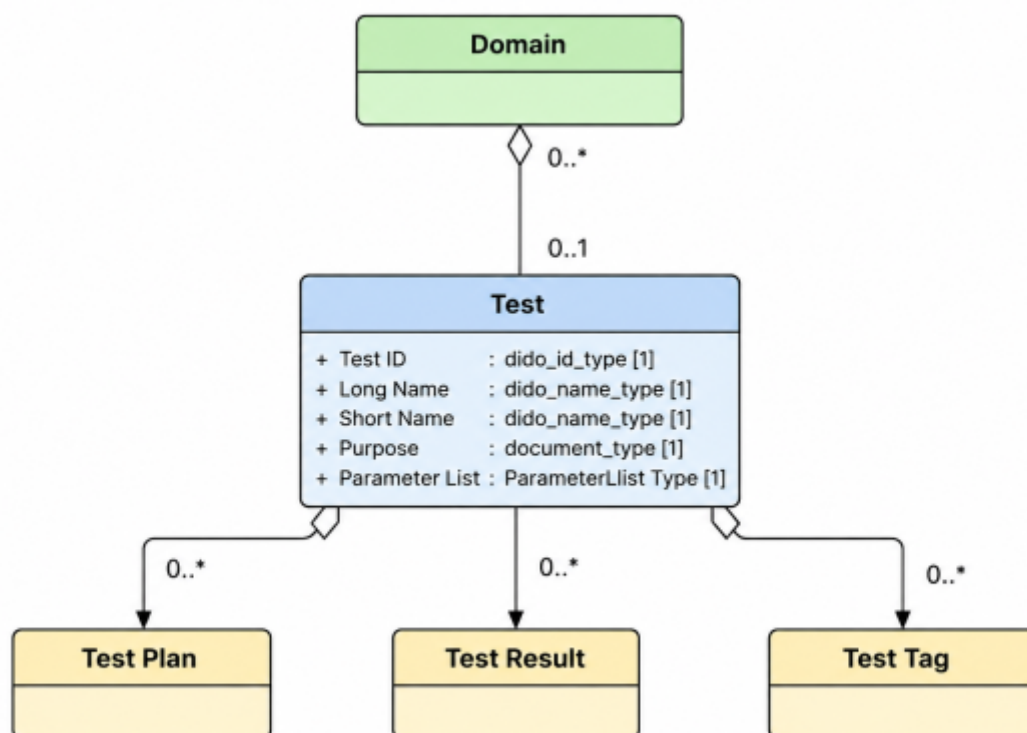


Figure 6.2-1: Test and its immediate source-model relationships

The diagram provides an overview of the immediate relationships represented by [DTE4]. Each relationship is described separately in the child pages of this section.

Source-Model Element

The [DTE4] class is named:

Test

The class resides at:

DIDO - RI
DIDO - Conceptual Model
Community
Testing Package
Test

Attributes

[DTE4] defines the following attributes and navigable properties for Test:

Attribute	Type	Multiplicity	Description
Test ID	dido_id_type	[1]	Identifier for the Test.
Long Name	dido_name_type	[1]	Long-form name of the Test.
Short Name	dido_name_type	[1]	Short-form name of the Test.
Purpose	document_type	[1]	Descriptive information identifying the purpose of the Test.
Parameter List	ParameterLlist Type	[1]	Parameter information associated with the Test.
fk_Test tagged wit Test Tags	Test_Tag	[0..1]	Source-model navigable property associated with Test tagging.
unnamed navigable property	Test_Result	[0..*]	Source-model property representing Test Results associated with the Test.

The attribute and property names above preserve the spelling and structure represented by [DTE4]. Apparent spelling or modeling issues are not silently corrected.

The reusable types referenced by these attributes are documented independently in [Section 5, Common Information Types](#).

Relationships to Other Classes

[DTE4] establishes direct relationships between Test and the following classes:

Related Class	Relationship
6.2.1 Domain	Associates a Test with its Domain context.
6.2.2 Test Plan	Provides subordinate organization of Test Sets and related testing structures.

Related Class	Relationship
6.2.3 Test Result	Associates Test Results with the Test.
6.2.4 Test Tag	Associates classification tags with the Test.

Each child page documents the [DTE4] multiplicities, architectural meaning, and any model-refinement considerations for that relationship.

Contents

- [6.2.1 Domain](#)
 - [6.2.2 Test Plan](#)
 - [6.2.3 Test Result](#)
 - [6.2.4 Test Tag](#)
-
- [6.2.1 Domain](#)

Parameter Structure

The Test element includes a required `Parameter List` property.

[DTE4] defines the type as:

ParameterLlist Type

The spelling `ParameterLlist` is preserved from [DTE4].

The Testing Package also defines supporting parameter structures, including:

- `Parameter_Type`;
- `Parameter_Definition_Type`;
- `Parameter_Diretion_Type`;
- `JSON String Type`; and
- `ParameterLlist Type`.

These reusable types belong to the Common Information Types portion of the DIDO-TE architecture and are documented independently in Section 5 rather than being defined by the Test class.

The extracted model indicates a potential structural issue: `ParameterLlist Type` contains values typed as `Parameter_Definition_Type` rather than `Parameter_Type`.

If the intent was to represent a list of parameter instances containing type, direction, required status, multiplicity, and default value, this relationship requires review.

DIDO-TE does not silently correct the [DTE4] structure.

Architectural Meaning

Within [DTE4], Test functions as the highest-level testing element represented by the Testing Package.

A Test provides identity and purpose for an organized testing activity and provides the source-model context for subordinate Test Plans, associated Test Results, and Test Tags.

The Test element therefore serves primarily as an organizing and identifying construct.

[DTE4] does not, by itself, establish all governed information required to define, configure, execute, evaluate, and provide evidence for a test within DIDO-TE.

Relationship to Test Definition

The [Test Definition](#) is a DIDO-TE architectural concept that is not represented explicitly as a class in [DTE4].

At this stage of the Information Model review:

- **Test** remains the [DTE4] source-model element;
- Test Definition remains the governed DIDO-TE concept; and
- no equivalence between Test and Test Definition is asserted.

The eventual relationship between Test and Test Definition is determined after the Test Plan, Test Set, Test Case, Test Step, Test Executable, Expected Results, Test Result, and related model elements have been reviewed.

Model Refinement Considerations

Review of the Test element raises the following model-refinement questions:

- Should **Test** remain a first-class DIDO-TE information element?
- Should **Test Plan** remain distinct from Test Definition?
- Is the Parameter List associated directly with Test required in the normalized DIDO-TE Information Model?
- Should every Test be associated with exactly one Domain?
- How should Test identity and versioning relate to the governed Test Definition?
- How should repeated [Test Executions](#) relate to Test?
- Should Test Results relate directly to Test, or should the relationship be mediated through Test Execution or Test Run?
- How should Test, Test Definition, Test Execution, Test Result, [Validation](#), [Evidence](#), and [Traceability](#) be connected?

These questions are recorded for normalization of the DIDO-TE Information Model and are not resolved by [DTE4].

Source

- [\[DTE4\] DIDO Reference Data Model](#), *DIDO - RI::DIDO - Conceptual Model::Community::Testing Package::Test*.

Notes for Editors

- **Test** is currently shown as a [DTE4] source-model class name and is not treated as a glossary-controlled DIDO-TE term.
- Document each direct Test relationship on its corresponding child leaf page.
- Define reusable datatypes, structured types, enumerations, identifiers, names, and parameter structures in [Section 5, Common Information Types](#) rather than on this page.
- Do not duplicate the complete semantics of a glossary-controlled class when the canonical glossary entry can be transcluded.
- Do not equate Test with Test Definition until the constituent testing elements have been reviewed.
- Preserve [DTE4] names, spellings, attributes, relationships, and multiplicities when documenting the source model.
- Record deliberate DIDO-TE corrections or refinements separately from the [DTE4] representation.
- Review the ParameterLlist Type structure before adopting it into the normalized DIDO-TE Information Model.
- Review whether the optional Domain relationship remains appropriate for DIDO-TE.

© 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/06-02-test/start>

Last update: **2026/08/23 17:27**

