

Sequenced Test Object

[Go up to Terms and Definitions](#)

Discussion

A Sequenced Test Object is a [Test Object](#) that contains or references a [Test Sequence](#) defining the progression of related test actions or validations.

OMG TestIF defines four specializations of Sequenced Test Object:

- [Test Set](#)
- [Test Case](#)
- [Test Step](#)
- [Test Executable](#)

A Sequenced Test Object can identify:

- The [Test Sequence](#)
- The [Sequence Steps](#) within the Test Sequence
- The first Sequence Step
- The progression among Sequence Steps
- Parallel execution paths
- Conditional execution paths
- Repeated execution paths
- Related [Test Items](#)
- Applicable [Test Environments](#)
- Applicable Test Arguments
- Applicable Test Argument Values
- Applicable [Expected Results](#)
- Relationships to other [Test Objects](#)
- Applicable [Governance Policies](#)
- [Provenance](#)
- [Traceability](#)

A Test Sequence uses [Sequence Steps](#) to reference the Sequenced Test Objects performed at specific positions in the execution progression.

This separation permits a reusable Sequenced Test Object to participate in multiple Test Sequences. Each Sequence Step can supply the Test Argument Values and Expected Results applicable to that occurrence.

Definition

[Test Object](#) that contains or references a [Test Sequence](#) defining the progression of related test actions or validations

Source

Adapted from:

- [OMG Test Information Interchange Format \(TestIF\), Version 1.0 Beta 3, Section 7.3.3, SequencedTestObject](#)
- [Test Object](#)
- [Test Sequence](#)
- [Sequence Step](#)
- [Test Environment](#)
- [Test Item](#)
- DIDO Reference Implementation Conceptual Model
- DIDO-TE draft Requirements Register

OMG TestIF defines SequencedTestObject as an abstract TestObject containing sequential procedural steps represented through one or more related TestSequence objects. This definition preserves its abstract parent role and its relationship to Test Sequence.

Note

Sequenced Test Object is an abstract concept. A concrete instance is classified as a:

- [Test Set](#)
- [Test Case](#)
- [Test Step](#)
- [Test Executable](#)

A Sequenced Test Object differs from a [Sequence Step](#):

- A Sequenced Test Object identifies reusable test information
- A Sequence Step identifies a position within a Test Sequence at which a Sequenced Test Object occurs

A Sequenced Test Object differs from a [Test Sequence](#):

- A Sequenced Test Object identifies the Test Object participating in execution
- A Test Sequence identifies the order and progression of Sequence Steps

A Sequenced Test Object can participate in more than one Test Sequence. Each occurrence can use different Test Argument Values or Expected Results when the applicable Test Definition permits them.

A Sequenced Test Object can reference multiple [Test Items](#) and [Test Environments](#).

A [Test Result](#) can reference the Sequenced Test Object or the Sequence Step that produced the recorded outcome.

A [Test Run](#) preserves the execution context and Test Results associated with the executed Sequenced Test Objects.

Example

A [Test Case](#) verifies a proposed DIDO state transition.

The Test Case is a Sequenced Test Object containing a [Test Sequence](#) with the following [Sequence Steps](#):

- Submit the proposed transaction
- Observe distribution of the transaction to the applicable [Nodes](#)
- Validate the proposed state transition
- Observe the resulting [DIDO](#) state
- Compare the observed state with the [Expected Result](#)

Each Sequence Step references the applicable [Test Step](#) or [Test Executable](#).

During [Test Execution](#), the resulting [Test Run](#) records the [Test Results](#) associated with the executed Sequence Steps and Sequenced Test Objects.

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

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

Permanent link:
https://wiki.didosolutions.com/dido/99_annexes/annex-b-terms-and-definitions/s/sequenced_test_object

Last update: **2026/08/04 12:44**

