

Test Executable

[Go up to Terms and Definitions](#)

Discussion

A Test Executable contains or references executable instructions that implement a test action or validation.

As a [Sequenced Test Object](#), a Test Executable can participate in a [Test Sequence](#).

A Test Executable can implement an action or validation specified by a [Test Step](#).

Executable instructions can include:

- Source code
- Compiled code
- Scripts
- Commands
- Declarative automation
- Tool-specific procedures
- Test-framework instructions
- Executable models
- Service requests
- Hardware-control instructions
- References to externally maintained executable artifacts

A Test Executable can identify:

- Its identity
- Its name
- Its description
- Its version
- Its executable instructions
- Its executable type
- Its language or format
- Its entry point
- Its applicable [Test Arguments](#)
- Its applicable [Test Argument Values](#)
- Its produced outputs
- Its applicable [Expected Results](#)
- Its required dependencies
- Its required execution facilities
- Its applicable [Test Items](#)
- Its applicable [Test Environments](#)
- Its completion condition
- Its failure condition
- Its timeout

- Its applicable [Governance Policies](#)
- Its [Provenance](#)
- Its [Traceability](#)

A [Sequence Step](#) identifies the occurrence of a Test Executable at a particular position within a [Test Sequence](#).

The same Test Executable can participate in multiple [Test Sequences](#) or occur more than once within the same Test Sequence.

Each Sequence Step can supply the applicable [Test Argument Values](#) and [Expected Results](#) for that occurrence.

Definition

[Sequenced Test Object](#) that contains or references executable instructions implementing a test action or validation

Source

Adapted from:

- [OMG Test Information Interchange Format \(TestIF\), Version 1.0 Beta 3, Section 7.3.7, TestExecutable](#)
- [Test Object](#)
- [Sequenced Test Object](#)
- [Test Step](#)
- [Test Sequence](#)
- [Sequence Step](#)
- [Test Argument](#)
- [Test Argument Value](#)
- [DIDO Reference Implementation Conceptual Model](#)
- [DIDO-TE draft Requirements Register](#)

OMG TestIF defines TestExecutable as a SequencedTestObject used for low-level procedural definitions, including code and scripts intended for test automation.

This definition preserves that automation role while allowing the executable information to reside within the Test Executable or in an independently managed artifact referenced by it.

Note

Every Test Executable is a [Sequenced Test Object](#) and therefore a [Test Object](#).

A Test Executable differs from a [Test Step](#):

- A [Test Step](#) specifies an action or validation independently of a particular execution mechanism
- A Test Executable contains or references the executable instructions implementing an action or validation

A Test Step can have:

- No Test Executable when a human or external process performs the Test Step
- One Test Executable when one executable artifact implements the Test Step
- Multiple Test Executables when different tools, platforms, or [Test Environments](#) require different realizations

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

- A Test Executable identifies reusable executable instructions
- A [Sequence Step](#) identifies an occurrence of a [Sequenced Test Object](#) within a [Test Sequence](#)

A Test Executable differs from a [Test Result](#):

- A Test Executable identifies the executable instructions used during [Test Execution](#)
- A [Test Result](#) records an outcome produced by Test Execution

A Test Executable can define [Test Arguments](#) representing its input and output interface.

A Test Executable does not assign occurrence-specific values to those Test Arguments. A [Sequence Step](#) supplies the applicable [Test Argument Values](#) for a particular occurrence of the Test Executable within a [Test Sequence](#).

A Test Executable can depend on a particular tool, programming language, operating system, processor architecture, or execution facility. The associated [Test Step](#) should remain independent of those implementation details when the action or validation can be expressed independently.

A change to the executable instructions, referenced artifact, dependency, entry point, [Test Argument](#) interface, or execution requirements can create a new Test Executable version.

A reproducible [Test Run](#) should identify:

- The exact Test Executable identity and version
- The referenced executable artifact
- The executable artifact digest
- The applicable [Test Arguments](#)
- The applicable [Test Argument Values](#)
- Every executable dependency used during [Test Execution](#)

Example

A [Test Case](#) contains a [Test Step](#) requiring submission of a proposed [DIDO](#) transaction.

A Test Executable references a versioned script that:

- Connects to the applicable [Node](#)
- Submits the proposed transaction

- Records the transaction identifier
- Returns the submission status

The Test Executable defines:

- transaction as a required input [Test Argument](#)
- targetNode as a required input [Test Argument](#)
- submissionIdentifier as an output [Test Argument](#)
- submissionStatus as an output [Test Argument](#)

A [Sequence Step](#) references the Test Executable and supplies:

- The proposed transaction as the [Test Argument Value](#) for transaction
- The applicable [Node](#) identity as the Test Argument Value for targetNode

During [Test Execution](#), the [Test Run](#) records:

- The Test Executable identity and version
- The referenced executable artifact
- The executable artifact digest
- The input [Test Argument Values](#)
- The output [Test Argument Values](#)
- The applicable [Test Environment](#)
- The resulting [Test Result](#)
- The supporting [Evidence](#)

© 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/t/test_executable

Last update: **2026/08/04 13:01**

