

Execution Facility

[Go up to Terms and Definitions](#)

Discussion

An Execution Facility is an [Operational Resource](#) that interprets or performs [Executable Instructions](#).

An Execution Facility can provide one or more capabilities required to execute instructions, including:

- Processing
- Interpretation
- Scheduling
- Automation
- Communication
- Data access
- State management
- Model execution
- Service invocation
- Device control
- Measurement
- Observation
- Result capture
- Error reporting

An Execution Facility can include:

- A processor
- A physical machine
- A virtual machine
- A container runtime
- A process
- A command processor
- An interpreter
- A compiler runtime
- An automation engine
- A workflow engine
- A test framework
- A model-execution engine
- A database engine
- A service
- A device controller
- A hardware test instrument
- A human-operated execution tool
- A combination of hardware, software, and services

An Execution Facility can identify:

- Its identity
- Its name
- Its description
- Its type
- Its [Version](#)
- Its supported instruction languages or formats
- Its supported operations
- Its available interfaces
- Its applicable [Endpoints](#)
- Its required dependencies
- Its resource requirements
- Its operating constraints
- Its supported [Executable Instructions](#)
- Its applicable [Executable Artifacts](#)
- Its applicable [Test Environments](#)
- Its applicable [Deployment Targets](#)
- Its applicable [Governance Policies](#)
- Its [Provenance](#)
- Its [Traceability](#)

A [Test Executable](#) contains or references the Executable Instructions used during [Test Execution](#).

The Execution Facility interprets or performs those Executable Instructions within the applicable [Test Environment](#).

A Test Environment can contain, provide, identify, or provide access to one or more Execution Facilities.

Definition

[Operational Resource](#) that interprets or performs [Executable Instructions](#)

Source

Adapted from:

- [OMG Test Information Interchange Format \(TestIF\), Version 1.0 Beta 3, Section 7.3.7, TestExecutable](#)
- [Executable Instruction](#)
- [Test Executable](#)
- [Test Execution](#)
- [Test Environment](#)
- DIDO Reference Implementation Conceptual Model
- DIDO-TE draft Requirements Register

OMG TestIF describes TestExecutable as a low-level procedural definition that supports test automation through executable information such as code and scripts.

OMG TestIF does not define Execution Facility as a separate TestIF model element. This definition introduces Execution Facility to identify the Operational Resource that interprets or performs the Executable Instructions associated with a Test Executable.

Note

An Execution Facility differs from an [Executable Instruction](#):

- An Executable Instruction specifies an operation
- An Execution Facility interprets or performs the operation

An Execution Facility differs from a [Test Executable](#):

- A Test Executable contains, organizes, or references Executable Instructions
- An Execution Facility interprets or performs those Executable Instructions

An Execution Facility differs from an [Executable Artifact](#):

- An Executable Artifact stores, packages, or provides Executable Instructions
- An Execution Facility loads, interprets, invokes, or performs those instructions

An Execution Facility differs from a [Test Environment](#):

- A Test Environment identifies the conditions, resources, configurations, and dependencies required to conduct testing
- An Execution Facility provides a capability for interpreting or performing Executable Instructions within that Test Environment

An Execution Facility differs from a [Deployment Target](#):

- A Deployment Target identifies a destination onto which a deployable artifact can be installed, configured, or activated
- An Execution Facility interprets or performs Executable Instructions

A Deployment Target can provide or host an Execution Facility. The same resource can act as both when it satisfies both definitions.

An Execution Facility differs from a [Node](#):

- A Node is an identifiable participant that performs one or more assigned roles in a distributed system
- An Execution Facility provides the capability to interpret or perform Executable Instructions

A Node can host, use, or provide access to one or more Execution Facilities. An Execution Facility does not become a Node unless it also acts as an identifiable participant with assigned roles in the distributed system.

An Execution Facility can use other Execution Facilities. For example, a test framework can invoke a command processor, which invokes a runtime, which sends a request to a service.

An Execution Facility can operate locally or remotely. Physical location does not determine whether a

resource qualifies as an Execution Facility.

An Execution Facility is not necessarily dedicated to testing. The facility qualifies when it interprets or performs Executable Instructions used for the applicable purpose.

A human does not ordinarily constitute an Execution Facility. A human-operated tool can constitute an Execution Facility when the tool interprets or performs the Executable Instructions and the human controls or initiates its operation.

A reproducible [Test Run](#) should identify the applicable Execution Facilities and their Versions, configurations, dependencies, and operating conditions.

Example

A [Test Executable](#) references a versioned script that submits a proposed transaction to a [Node](#).

The applicable Execution Facilities include:

- A container runtime that starts the test container
- A command processor that invokes the script
- A language interpreter that interprets the script
- A test framework that supplies the applicable [Test Argument Values](#)
- A service that processes the resulting request

During [Test Execution](#), the Execution Facilities:

- Load the referenced [Executable Artifact](#)
- Interpret or perform its Executable Instructions
- Submit the proposed transaction through the applicable [Endpoint](#)
- Capture the response
- Provide the resulting data to the [Test Run](#)

The Test Run records:

- The identity of each Execution Facility
- The Version of each Execution Facility
- The applicable configuration
- The applicable dependencies
- The applicable [Test Environment](#)
- The resulting [Test Result](#)
- The supporting [Evidence](#)

From:

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

Permanent link:

https://wiki.didosolutions.com/dido/99_annexes/annex-b-terms-and-definitions/e/executable_artifact

Last update: **2026/08/04 13:22**

