Executable Instruction

Go up to Terms and Definitions

Discussion

An Executable Instruction specifies an operation that an Execution Facility can interpret or perform.

An Executable Instruction can implement an action, calculation, validation, communication, state transition, measurement, observation, or control.

Executable Instructions can take the form of:

- Source code
- Compiled code
- Scripts
- Commands
- Declarative automation
- Tool-specific procedures
- Test-framework instructions
- Executable models
- Service requests
- Hardware-control instructions

An Executable Instruction can identify:

- Its identity
- Its name
- Its description
- Its Version
- Its instruction type
- Its language or format
- Its entry point
- Its applicable Test Arguments
- Its applicable Test Argument Values
- Its required inputs
- Its produced outputs
- 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 Test Executable contains or references one or more Executable Instructions that implement a test action or validation.

The Test Executable can contain the Executable Instructions directly or reference an independently maintained Executable Artifact that contains them.

An Execution Facility interprets or performs the Executable Instructions during Test Execution.

Definition

instruction that an Execution Facility can interpret or perform to implement an action, calculation, validation, or control

Source

Adapted from:

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

OMG TestIF describes a TestExecutable as a low-level procedural definition that can contain code or scripts for test automation.

OMG TestIF does not define Executable Instruction as a separate TestIF model element. This definition introduces the term to distinguish the individual instructions from the Test Executable that contains or references them and from the artifact in which they can reside.

Note

An Executable Instruction differs from a Test Step:

- A Test Step specifies an action or validation independently of a particular execution mechanism
- An Executable Instruction provides part or all of the machine-interpretable or tool-interpretable realization of that action or validation

An Executable Instruction differs from a Test Executable:

- An Executable Instruction specifies an individual executable operation
- A Test Executable organizes, contains, or references one or more Executable Instructions

An Executable Instruction differs from an Executable Artifact:

- An Executable Instruction specifies an operation
- An Executable Artifact stores, packages, or otherwise provides one or more Executable Instructions

A Test Executable can reference an Executable Artifact without copying the Executable Instructions into the Test Executable.

An Executable Instruction does not require direct execution by a processor. An interpreter, command processor, automation engine, test framework, model-execution engine, service, device controller, or other Execution Facility can interpret or perform the instruction.

A natural-language direction intended only for human interpretation is not necessarily an Executable Instruction. The direction qualifies as an Executable Instruction when an identified Execution Facility can interpret or perform it according to defined execution rules.

An Executable Instruction is not necessarily:

- Source code
- Compiled code
- Independently deployable
- Independently reusable
- Specific to testing
- Contained within the Test Executable
- Stored in a particular file
- Associated with a particular programming language

A change to the operation, parameters, control flow, dependencies, expected behavior, or execution requirements can create a new Version of an Executable Instruction under the applicable versioning policy.

A reproducible Test Run should identify the exact Versions of the Executable Instructions or referenced Executable Artifacts used during Test Execution.

Example

A Test Step requires a Node to submit a proposed transaction to a specified Communication Endpoint.

A Test Executable references a versioned script containing Executable Instructions that:

- Read the applicable Test Argument Values
- Establish a connection to the Communication Endpoint
- Submit the proposed transaction
- Capture the response
- Record the transaction identifier
- Return the submission status
- Preserve the resulting Evidence

During Test Execution, the applicable Execution Facility interprets or performs the Executable Instructions.

The Test Run records:

- The Test Executable identity
- The Test Executable Version
- The referenced Executable Artifact
- The Executable Artifact Version
- The Executable Artifact digest
- The applicable Test Argument Values
- The applicable Test Environment
- The resulting Test Result
- The supporting Evidence
