

Version

[Go up to Terms and Definitions](#)

Discussion

A Version identifies a distinguishable state of an [Artifact](#).

An Artifact can have multiple Versions when its content, structure, behavior, configuration, interface, dependencies, or other governed characteristics change.

A Version can identify the state of an Artifact such as a:

- Document
- Specification
- Requirement
- [Data Object](#)
- Data model
- Schema
- Configuration
- Policy
- Interface
- Software component
- Executable artifact
- [Test Object](#)
- [Test Definition](#)
- [Test Executable](#)
- [Test Environment](#)
- [Baseline](#)

A Version should have a Version Identifier that distinguishes it from other Versions of the same Artifact.

A Version can also identify:

- The Artifact to which it applies
- Its Version Identifier
- Its creation time
- Its creator or responsible authority
- Its relationship to preceding and succeeding Versions
- The changes represented by the Version
- Its applicable status
- Its applicable dependencies
- Its content digest
- Its applicable [Governance Policies](#)
- Its [Provenance](#)
- Its [Traceability](#)

The applicable versioning policy determines:

- Which changes require a new Version
- How Version Identifiers are assigned
- Whether Version Identifiers imply an ordering
- Whether a Version can be modified after identification
- How Versions relate to releases
- How obsolete or superseded Versions are treated
- How compatibility among Versions is represented

[Version Control](#) manages distinguishable Versions and the relationships among them.

A [Version Control System](#) provides mechanisms for performing Version Control.

Definition

identified state of an [Artifact](#) distinguished from other states of that Artifact

Source

Adapted from:

- [Version Control](#)
- [Version Control System](#)
- DIDO Reference Implementation Conceptual Model
- DIDO-TE draft Requirements Register

This definition establishes Version as the identified state managed through Version Control. It separates the Version itself from the process and system used to manage it.

Note

A Version is distinct from [Version Control](#):

- A Version identifies a distinguishable state of an Artifact
- Version Control is the process for managing distinguishable revisions and the relationships among them

A Version is distinct from a [Version Control System](#):

- A Version identifies the state being managed
- A Version Control System provides the mechanisms used to manage Versions

A Version is not necessarily a [Baseline](#). A Version becomes part of a Baseline only when the applicable authority selects and governs it as part of that Baseline.

A Version is not necessarily:

- Approved
- Accepted
- Released
- Published
- Deployed
- Operational
- Current
- Compatible with another Version

Those characteristics require separate status, governance, release, deployment, or compatibility information.

A Version Identifier does not necessarily indicate chronological order, precedence, maturity, approval status, or compatibility unless the applicable versioning policy defines that meaning.

Two copies of an Artifact can represent the same Version when they preserve the same identified state. Conversely, two Artifacts with similar names do not represent the same Version unless their identities and governed states establish that relationship.

A change in storage location, packaging, encoding, or representation does not necessarily create a new Version when the governed content and semantics remain unchanged. The applicable versioning policy determines whether such a change requires a new Version.

A reproducible [Test Run](#) should identify the exact Versions of the applicable Test Objects, Test Executables, Test Environments, configurations, policies, dependencies, and other governed Artifacts.

Example

A [Test Executable](#) has the Version Identifier 2.3.0.

That Version identifies:

- The executable instructions
- The applicable [Test Arguments](#)
- The required dependencies
- The entry point
- The execution requirements
- The executable artifact digest
- The applicable [Test Environments](#)

A later change to the executable instructions creates Version 2.3.1 under the applicable versioning policy.

A [Test Run](#) identifies Version 2.3.0 and its executable artifact digest. This information distinguishes the Test Executable used during the Test Run from Version 2.3.1 and supports [Provenance](#), [Traceability](#), and reproducibility.

© 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/v/version

Last update: **2026/08/04 13:10**

