

Data Object

[Go up to Terms and Definitions](#)

Discussion

A Data Object encapsulates data as an identifiable instance of a defined type.

The identity of a Data Object distinguishes the Data Object from other instances of the same type. The defined type establishes the characteristics, structure, constraints, or permissible operations associated with the Data Object.

A Data Object can represent:

- A real-world entity
- A conceptual entity
- An event
- A transaction
- A message
- A document
- A state
- A state transition
- A configuration
- A test
- A test result
- An Evidence record
- A relationship among other Data Objects

A Data Object remains conceptually distinct from its physical representation. Different implementations can represent the same Data Object using different serialization formats, storage mechanisms, programming-language objects, database records, messages, files, or memory structures.

A Data Object can have:

- An identity
- A type
- A state
- A lifecycle
- A version
- A predecessor
- A successor
- Relationships to other Data Objects
- Provenance
- Traceability
- Applicable constraints
- Permitted operations

The applicable architecture determines which characteristics a particular Data Object requires.

Definition

identifiable information object that encapsulates data as an instance of a defined type

Source

Adapted from:

- [DIDO Reference Architecture, Data Object](#)
- [DIDO CLI, What is DIDO?](#)
- Object-oriented modeling and information-modeling usage
- [Immutable Data Object](#)
- [Distributed Immutable Data Object \(DIDO\)](#)

The definition preserves the DIDO-RA treatment of a Data Object as an instance of a class used within or distributed across a system while remaining independent of a specific modeling or implementation technology.

Note

A Data Object is not necessarily immutable or distributed.

An [Immutable Data Object](#) preserves each committed state without in-place alteration.

A [Distributed Immutable Data Object \(DIDO\)](#) distributes the committed states of an Immutable Data Object across multiple networked Nodes through a shared state-agreement mechanism.

A Data Object is distinct from:

- The real-world or conceptual subject it represents
- The type that defines its permitted characteristics
- A particular serialization of the Data Object
- A particular copy of the Data Object
- The storage mechanism containing the Data Object
- The message carrying the Data Object
- The software component processing the Data Object

Changing a physical representation does not create a different Data Object when the identity, type, state, and applicable semantics remain unchanged.

Creating a state with a new identity, version, or lifecycle position can create a successor Data Object when the governing model defines each state as a separately identifiable object.

Example

A trade model defines a Trade type.

A particular trade identified as TRADE - 2026 - 0042 constitutes a Data Object that instantiates the Trade type. The trade can appear as a database record, a serialized message, a document, or an in-memory representation without changing its conceptual identity.

If the architecture preserves each committed trade state as a separately identifiable successor, each state can constitute an [Immutable Data Object](#).

© 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/d/data_object

Last update: **2026/08/04 06:58**

