

Liveliness

[Return to Quality of Service](#)

The **LIVELINESS** policy controls the mechanism and parameters used by the Service to ensure that particular entities on the network are still “alive.” The liveliness can also affect the ownership of a particular **instance**, as determined by the **OWNERSHIP QoS** policy.

This policy has several settings to support both data-objects that are updated periodically as well as those that are changed sporadically. It also allows customizing for different application requirements in terms of the kinds of failures that will be detected by the liveliness mechanism.

The **AUTOMATIC** liveliness setting is most appropriate for applications that only need to detect failures at the processlevel, but not application-logic failures within a process. The Service takes responsibility for renewing the leases at the required rates and thus, as long as the local process where a **DomainParticipant** is running and the link connecting it to remote participants remains connected, the entities within the **DomainParticipant** will be considered alive. This requires the lowest overhead.

The **MANUAL** settings (**MANUAL_BY_PARTICIPANT**, **MANUAL_BY_TOPIC**), require the application on the publishing side to periodically assert the liveliness before the lease expires to indicate the corresponding **Entity** is still alive. The action can be explicit by calling the `assert_liveliness` operations, or implicit by writing some data.

The two possible manual settings control the granularity at which the application must assert liveliness.

- The setting **MANUAL_BY_PARTICIPANT** requires only that one Entity within the **publisher** is asserted to be alive to deduce all other Entity objects within the same **DomainParticipant** are also alive.
- The setting **MANUAL_BY_TOPIC** requires that at least one instance within the **DataWriter** is asserted.

The value offered is considered compatible with the value requested if and only if the following conditions are met:

1. the inequality `offered kind >= requested kind` evaluates to **TRUE**. For the purposes of this inequality, the values of **LIVELINESS** kind are considered ordered such that: **AUTOMATIC** < **MANUAL_BY_PARTICIPANT** < **MANUAL_BY_TOPIC**.
2. the inequality `offered lease_duration ≤ requested lease_duration` evaluates to **TRUE**.

Changes in **LIVELINESS** must be detected by the Service with a time-granularity greater or equal to the `lease_duration`. This ensures that the value of the `LivelinessChangedStatus` is updated at least once during each `lease_duration` and the related Listeners and WaitSets are notified within a `lease_duration` from the time the **LIVELINESS** changed.

Source: [DDS 1.4 Spec](#)

Last
update: 2021/06/06 15:53 dido:public:ra:xapend:xapend.f_qos:11_liveliness https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.f_qos:11_liveliness&rev=1623009221

From:
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.f_qos:11_liveliness&rev=1623009221

Last update: **2021/06/06 15:53**

