

# TIME\_BASED\_FILTER QoS Parameter

[return to the Quality of Service \(QoS\) Parameters](#)

The TIME\_BASED\_FILTER policy allows a DataReader to indicate that it does not necessarily want to see all values of each [instance](#) published under the [Topic](#). Rather, it wants to see at most one change every `minimum_separation` period.

The TIME\_BASED\_FILTER applies to each instance separately, that is, the constraint is that the DataReader does not want to see more than one [sample](#) of each instance per `minimum_separation` period.

This setting allows a DataReader to further decouple itself from the DataWriter objects. It can be used to protect applications that are running on a heterogeneous network where some nodes are capable of generating data much faster than others can consume it. It also accommodates the fact that for fast-changing data different subscribers may have different requirements as to how frequently they need to be notified of the most current values.

The setting of a TIME\_BASED\_FILTER, that is, the selection of a `minimum_separation` with a value greater than zero is compatible with all settings of the [HISTORY](#) and [RELIABILITY QoS](#). The TIME\_BASED\_FILTER specifies the samples that are of interest to the DataReader. The HISTORY and RELIABILITY QoS affect the behavior of the [middleware](#) with respect to the samples that have been determined to be of interest to the DataReader, that is, they apply after the TIME\_BASED\_FILTER has been applied.

In the case where the reliability QoS kind is RELIABLE then in steady-state, defined as the situation where the DataWriter does not write new samples for a period “long” compared to the `minimum_separation`, the system should guarantee delivery the last sample to the DataReader.

The setting of the TIME\_BASED\_FILTER `minimum_separation` must be consistent with the [DEADLINE](#) period. For these two QoS policies to be consistent they must verify that `period >= minimum_separation`. An attempt to set these policies in an inconsistent manner when an [entity](#) is created or via a `set_qos` operation will cause the operation to fail.

Source: [DDS 1.4 Spec](#)

From:  
<https://omgwiki.org/ddsf/> - **DDS Foundation Wiki**

Permanent link:  
[https://omgwiki.org/ddsf/doku.php?id=ddsf:public:guidebook:06\\_append:02\\_quality\\_of\\_service:time\\_based\\_filter](https://omgwiki.org/ddsf/doku.php?id=ddsf:public:guidebook:06_append:02_quality_of_service:time_based_filter)

Last update: **2021/07/14 15:55**

