

F.21 Writer Data Lifecycle

[Return to DDS Quality of Service](#)

The `WRITER_DATA_LIFECYCLE` policy controls the behavior of the `DataWriter` with regards to the lifecycle of the data-[instances](#) it manages, that is, the data-instances that have been either explicitly registered with the `DataWriter` using the register operations or implicitly by directly writing the data.

The `autodispose_unregistered_instances` flag controls the behavior when the `DataWriter` unregisters an instance by means of the `unregister` operations:

- The setting **`autodispose_unregistered_instances = TRUE`** causes the `DataWriter` to dispose the instance each time it is unregistered. The behavior is identical to explicitly calling one of the `dispose` operations on the instance prior to calling the `unregister` operation.
- The setting **`autodispose_unregistered_instances = FALSE`** will not cause this automatic disposition upon unregistering. The application can still call one of the `dispose` operations prior to unregistering the instance and accomplish the same effect. There are consequences of disposing and unregistering instances.

Note

the deletion of a `DataWriter` automatically unregisters all data-instances it manages. Therefore the setting of the `autodispose_unregistered_instances` flag will determine whether instances are ultimately disposed when the `DataWriter` is deleted either directly by means of the `Publisher::delete_datawriter` operation or indirectly as a consequence of calling `delete_contained_entities` on the `Publisher` or the `DomainParticipant` that contains the `DataWriter`.

Source: [DDS 1.4 Spec](#)

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:writer_data_lifecycle&rev=1624587441

Last update: **2021/06/24 22:17**

