부록: OMG QoS 요약

|  |  |  |  |
| --- | --- | --- | --- |
| QoS 명칭 | 설명 | RxO[[1]](#footnote-1) | 대상  엔티티[[2]](#footnote-2) |
| **HistoryQosPolicy** | 송신할 데이터의 크기와 저장된 데이터의 관리 방법을 제어할 수 있다. | No | T,R,W |
| **LifespanQosPolicyLifespanQosPolicy** | 어플리케이션이 전송하는 데이터를 얼마 동안 유효한 상태로 유지할 것인지 설정한다. | N/A | T,W |
| **OwnershipQosPolicy** | 복수의 DataWriter가 전송하는 샘플들을 동시에 수신했을 때 DataReader의 수신 방법을 지정할 수 있다. 이 설정이 “shared”인 경우 DataReader는 매칭되는 DataWriter가 수신하는 모든 토픽을 수신하게 된다. 만약 이 설정을 “exclusive”로 변경하는 경우 DataReader는 한번에 하나의 DataWriter가 전송하는 토픽만 수신하게된다. | Yes | T,R,W |
| **OwnershipStrengthQosPolicy** | 복수의 DataWriter가 전송하는 인스턴스의 강도(strength) 값을 설정할 수 있다. **OwnershipQosPolicy** 이 “exclusive”의 경우 이 값이 높은 샘플에 대해서 DataReader는 수신한다. | N/A | W |
| **ReaderDataLifecycleQosPolicy** | 수신된 데이터를 어떻게 관리할 것인지 DataReader 를 설정할 때 사용된다. 일반적으로 DataReader는 데이터를 수신하고 이를 큐에 보관한다. 어플리케이션은 이 큐에 저장된 데이터를 읽게 된다. 이 QoS를 이용하여 큐내의 특정 데이터를 제거할 수 있다. 예를 들어, DataWriter가 더 이상 살아 있지 않은 데이터를 큐에서 자동으로 삭제할 수 있다. | N/A | R |
| **WriterDataLifecycleQosPolicy** | 이 QoS는 키를 갖는 토픽의 인스턴스에 적용된다. DataWriter가 인스턴스의 등록을 해제하는 경우 해당 인스턴스를 모두 제거하는 것이 DataWriter의 기본 동작이다.  토픽의 소유권이 독점(exclusive)적인 경우, DataWriter는 이 토픽의 특정 인스턴스의 소유권을 양도하여 다른 DataWriter가 **OwnershipStrengthQosPolicy** 의 값과는 상관없이 데이터를 업데이트할 수 있도록 한다. | N/A | W |
| **PresentationQosPolicy** | 특정 데이터 셋의 일부가 아니라 전부를 한번에 수신하고자 하는 등을 조절하고자 할 때 사용된다. 예를 들어 키가 있는 데이터 샘플의 경우 동일한 키를 갖는 데이터를 한꺼번에 수신하고자 하는 경우 사용될 수 있다. | Yes | P,S |
| **TransportPriorityQosPolicy** | 일부 트랜스포트 프로토콜의 경우에 운영체제나 스위칭하드웨어에서 사용 가능한 사용자가 설정 가능한 “우선순위” 개념을 가지고 있다. 이런 경우에만 사용 가능한 QoS 이다. | N/A | T,W |
| **UserDataQosPolicy** | 이 QoS를 이용하여 디스커버리중에 부가적으로 보내고자 하는 데이터를 첨부할 수 있다. **UserDataQosPolicy** 는 DataReader/DataWriter 수준에서, **GroupDataQosPolicy** 는 Publishe/Subscriber 수준에서, **TopicDataQosPolicy** 는 토픽 수준에서 디스커버리 메타 데이터를 첨부한다. 이 QoS를 이용하여 보안 인증서를 디스커버리 시 보낼 수 있다. | No | D,R,W |
| **TopicDataQosPolicy** | T |
| **GroupDataQosPolicy** | P,S |
| **LatencyBudgetQosPolicy** | 데이터 전송에 걸리는 지연 시간을 설정한다. 보통의 경우에 0 이 사용된다. 0의 의미는 지연 시간을 최소화 하라는 의미이다. | Yes | T,R,W |
| **DeadlineQosPolicy** | 데이터 샘플간의 최대 시간 간격을 조절할 수 있다. DataReader의 경우 이 시간을 벗어난 샘플에 대해서는 DDS 미들웨어는 사용자 콜백을 호출한다. DataWriter의 경우 이 시간을 넘어서는 데이터는 전송하지 않는다. | Yes | T,R,W |
| **TimeBasedFilterQosPolicy** | 새로운 데이터의 최소 도착 시간을 설정한다. 최소 시간을 벗어다는 데이터는 전송하지 않는다. 이 QoS를 이용하여 너무 많은 데이터가 수신되지 않도록 조절할 수 있다. 예를 들어, GUI에 1초 마다 업데이트만 되는 데이터를 DataWriter가 너무 빠르게 보내는 경우, 이 QoS를 사용하여 전송 빈도를 조절할 수 있다. 동일한 토픽을 수신을 하는 다양한 DataReader에 대해서도 각각 이 QoS를 설정하여 데이터 수신 빈도를 조절할 수 있다. | N/A | R |
| **EntityFactoryQosPolicy** | DDS 엔티티의 행위를 제어할 수 있다. 예를 들어, DataReader가 enabled 상태로 생성되는 경우, DataReader는 생성 즉시 디스커버리가 수행되고, 즉시 데이터를 수신할 수 있게된다. 반대로 disabled 상태로 생성되는 경우 모든 초기화 절차가 완료되기 전에는 데이터 수신을 하지 못하도록 설정할 수 있다. | No | F,D,P,S |
| **PartitionQosPolicy** | 동일한 토픽에 대하여 DataWriter와 DataReader를 매칭하기 위한 string 식별자를 추가할 수 있다. 일반적으로 DataWriter와 DataReader는 토픽을 기준으로 매칭된다. 하지만 이 QoS가 적용되면 동일한 string을 갖는 Publisher와 Subscriber내의 DataWriter와 DataReader가 매칭되게 된다. 따라서 이 QoS를 이용하여 DataWriter와 DataReader의 매칭 범위를 임의로 조절할 수 있다. | No | P,S |
| **ReliabilityQosPolicy** | 네트워크에 의해서 분실된 샘플에 대하여 DDS 미들웨어가 재전송을 할 것인지 설정할 수 있다. 이 설정을 통하여 샘플이 항상 송신 순서대로 수신되도록 할 수 있다. BEST\_EFFORT로 설정되는 경우, DDS 미들웨어는 송수신 보장을 위하여 아무런 조치를 취하지 않으나. 이 설정의 장점은 가장 빠르며, 효율적이다. 주로 센서들이 주기적으로 보내는 상태 메시지에 많이 사용된다. 이 QoS와 더불어 **HistoryQosPolicy** 와 **ResourceLimitsQosPolicy** 사용함으로써 Reliable 수준을 정할 수 있다. 예를 들어, **HistoryQosPolicy** 의 설정이 KEEP\_ALL로 설정되는 경우 DataWriter는 모든 샘플이 수신될 때까지 내부 큐에 보관하며, DataReader는 어플리케이션이 데이터를 읽을 때까지 내부 큐에 보관하게된다. 이러한 내부 큐의 크기는 **ResourceLimitsQosPolicy** 를 이용하여 설정할 수 있다. | Yes | T,R,W |
| **LivelinessQosPolicy** | DataReader와 매칭되는 DataWriter가 네트워크와 단절되거나 삭제될 경우를 검출할 수 있는 방법을 제공한다. | Yes | T,R,W |
| **DurabilityQosPolicy** | 새롭게 또는 나중에 조인되는 DataReader에게 조인 이 전 시점의 데이터를 전송할 것인지를 지정할 수 있다. | Yes | T,R,W |
| **DurabilityServiceQosPolicy** | PERSISTENT 또는 TRANSIENT Durability QoS를 갖는 DataWriter가 외부 저장 서비스를 설정할 수 있다. 대부분의 경우 이 QoS는 **DurabilityQosPolicy** 와 함께 사용된다. | No | T,W |
| **ResourceLimitsQosPolicy** | 데이터 캐슁을 위하여 메모리 할당 크기를 조정할 수 있다. | No | T,R,W |
| **DestinationOrderQosPolicy** | 복수의 DataWriter가 전송하는 동일한 토픽에 대하여 어떻게 다룰 것인지 설정할 수 있다. “by source timestamp” 기준인 경우 DataReader는 DataWriter가 전송한 순서대로 데이터를 수신할 수 있으며, “by reception timestamp” 기준인 경우 DataReader는 수신된 시간 기준으로 데이터를 수신한다. 이는 UDP 프로토콜이 데이터의 도달 순서를 보증하지 않기 때문에 도착 순서를 변경할 때 유용한 기능이다. | Yes | T,R,W |

1. RxO 가 “Yes”인 경우 publishing측과 subscribing 측은 반듯이 QoS가 호환성이 있어야 함을 의미한다. “No”인 경우 QoS 가 publishing측과 subscribing 양측에 설정은 되어야 하나 그 값은 서로 독립적임을 의미한다. “N/A”인 경우 이 QoS 는 publishing측과 subscribing측 중 어느 한 쪽에만 정의되어 됨을 의미하며 호환성과 관계가 없음을 나타낸다. [↑](#footnote-ref-1)
2. F=DomainParticipantFactory, D=DomainParticipant, T=Topic, P=Publisher, W=DataWriter, S=Subscriber, R=DataReader [↑](#footnote-ref-2)