

RTI® Queuing Service

Fast, reliable and scalable load-balancing for Connex® DDS applications



Message queues provide a foundation for elastic distributed processing. They store messages and deliver each reliably to exactly one consumer. Message queues enable implementation of scalable and high-performance data analytics through load-balancing and efficient job allocation.

Highlights:

Implements one-to-one queuing communication pattern

Load balances publish-subscribe and request-reply interactions

Persists messages in memory or to disk

Supports fully redundant transactional messaging for high availability and fault tolerance

API for remote administration and monitoring

Dead-letter queue for analyzing message delivery failures

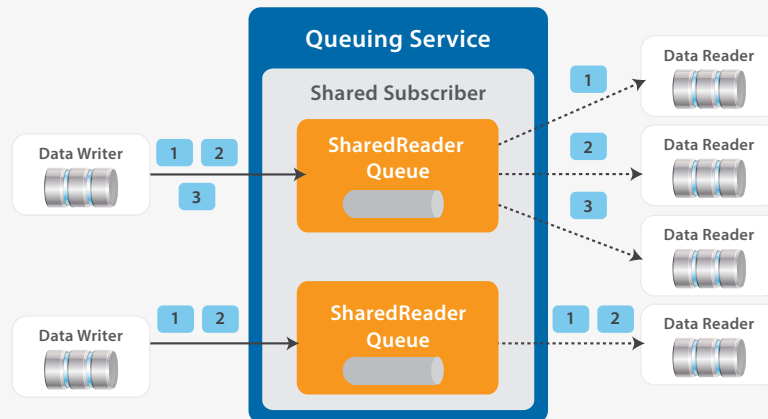
Works seamlessly with RTI infrastructure services and tools

Can replace other brokers including AMQP and RabbitMQ

The RTI Queuing Service enables point-to-point messaging in Connex DDS. It brokers interactions between message producers (DDS Data Writers) and consumers (DDS Data Readers), delivering each message (DDS sample) to only one consumer. This balances workloads by distributing jobs across a pool of processors to take advantage of elastic computing capabilities – whether in the cloud or at the edge. This takes distributed data processing to a new level of scalability with minimal end-to-end latency.

With the RTI Queuing Service, Connex DDS supports all the fundamental communication patterns used in the Industrial Internet of Things (IIoT): publish/subscribe for distributing streaming data and asynchronous events, request/reply for controlling and managing devices and queuing for scalable data processing and analytics. This eliminates the need for adopting and integrating different technologies for each of these system requirements. It also simplifies administration by minimizing the number of required infrastructure technologies.

RTI Queuing Service allows multiple consumers (Data Readers) to collaborate, coordinate and balance workload. The Data Readers can reside within the same process, different processes or even different machines.

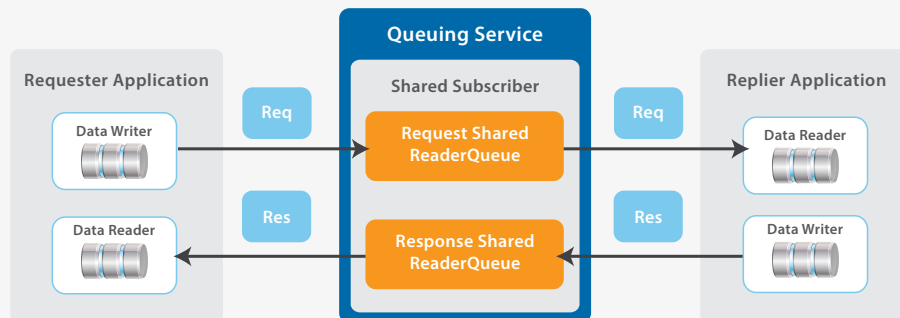


Each message is delivered to a single Data Reader in a round-robin manner. Based on the dispatch mode configured for a queue, Queuing Service will dispatch a message to a Data Reader that has explicitly indicated its availability, has acknowledged all its previous messages or is under a specified threshold of unacknowledged messages.

Queues and service configuration can be replicated across Queuing Service instances. A master-slave architecture guarantees availability of the Queuing Service at any time.

Queuing Service includes a REST-like API for remote administration. Supported operations include queue creation, deletion and introspection. Transactions can also be monitored by subscribing to the underlying publish/subscribe interactions.

Queuing Service supports a request-reply communication model. A Requester Application sends a sample to a Shared Reader Queue. A replier application receives the sample from the Shared Reader Queue and returns a response to the requester application.



About RTI

Real-Time Innovations (RTI) is the Industrial Internet of Things (IIoT) connectivity company. The RTI Connex[®] databus is a software framework that shares information in real time, making applications work together as one, integrated system. It connects across field, fog and cloud. Its reliability, security, performance and scalability are proven in the most demanding industrial systems. Deployed systems include medical devices and imaging; wind, hydro and solar power; autonomous planes, trains and cars; traffic control; Oil and Gas; robotics, ships and defense.

RTI is the largest vendor of products based on the Object Management Group (OMG) Data Distribution Service[™] (DDS) standard. RTI is privately held and headquartered in Sunnyvale, California.



Your systems. Working as one.

CORPORATE HEADQUARTERS
232 E. Java Drive
Sunnyvale, CA 94089
Tel: +1 (408) 990-7400
Fax: +1 (408) 990-7402
info@rti.com
www.rti.com