

## RTI Persistence Service

### Providing Reliable Data Availability

#### BENEFITS

- Improves data durability and fault tolerance
- Simplifies application design by further decoupling subscribers from publishers' lifecycles
- Increases scalability by enabling load balancing for publishers

RTI Persistence Service provides reliable data availability for RTI Data Distribution Service applications. It ensures that data from RTI Data Distribution Service applications is available where and when it's needed; that data can be recovered, transmitted or received as publishers and subscribers join or leave the system.

With RTI Persistence Service:

- Durable subscribers that go offline receive missed messages when they rejoin the system
- Messages may be stored in an in-memory and/or on-disk persistence store for the optimal balance of latency, data durability and history
- Subscribers may request a replay of messages, even if the original publisher is no longer available
- By using the persistence service as a "broker", publisher resources can be prioritized to those subscribers that require lowest latency
- Multiple persistence stores can run concurrently for redundancy and load balancing

#### How It Works

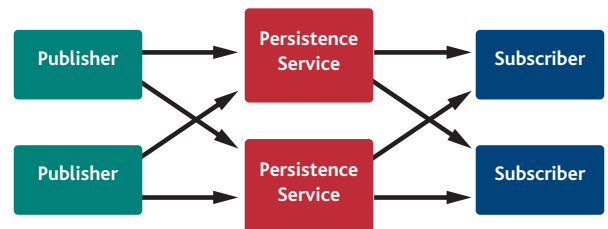
RTI Persistence Service runs as a separate RTI Data Distribution Service application; you can run it on the same node as the publishing application, the subscribing application, or an entirely different node in the network. If you choose to save the data in permanent storage, it is saved in a relational database.

RTI Persistence Service provides two ways to achieve reliable data availability: relay communication and direct communication:

#### RTI Persistence Service with Relay

**Communication:** publishers communicate with the persistence service, not directly with each other, and the service acts as a "broker" to deliver data to the subscribers. The subscribers still receive data even if the publisher terminates, and subscribers with the lowest latency requirements can be prioritized higher.

Any participants *not* configured to use RTI Persistence Service will still communicate directly with one another.



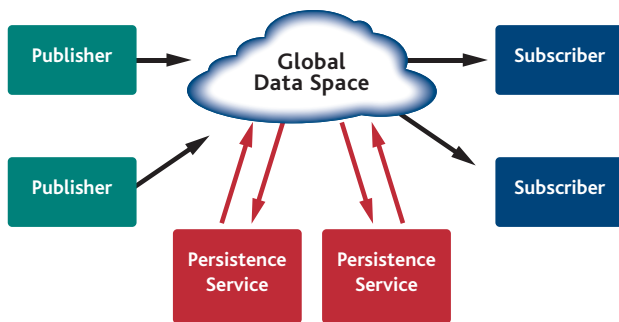
*Relay communication model allows RTI to act as a "broker" that prioritizes subscribers with the lowest latency requirements*

#### RTI Persistence Service with Direct

**Communication:** publishers and subscribers that are configured for reliable data availability communicate directly with each other, minimizing any overhead associated with data persistence. When the persistence service detects loss of a publisher, it mimics that publisher and sends its data to late-joining subscribers.

#### Advantages

**Data Durability:** Data durability is achieved by storing the previously published information in memory, or in persistent storage like a relational database. In this manner, a late-joining subscriber can receive the information even after the publishing application is no longer executing.



*Direct communication model minimizes performance overhead of data persistence*

**Durable Writer History:** A publishing application can now persist its local history cache so that it can survive shutdowns, crashes, and restarts. When the application restarts, each publisher automatically loads all the data in its history cache from the disk and can carry on sending data as if it had never stopped executing. To the rest of the system, it appears as if the publishing application was temporarily disconnected from the network and then re-appeared.

**Durable Reader State:** A subscribing application can now locally persist its state and remember the data it has already received. When the application restarts, each subscriber automatically loads its state from disk and can carry on receiving data as if it had never stopped executing. Data already received by the subscriber will not be resent.

**Fault Tolerance and Load Balancing:** By configuring *multiple* RTI Persistence Service instances to persist the same topics, users can achieve fault tolerance. By distributing the set of topics across several RTI Persistence Service instances, users can reap the benefits of load balancing.

### Specifications

RTI Persistence Service implements the optional Persistence Profile defined in the Object Management Group's (OMG) Data Distribution Service for Real-Time Systems specification.

RTI Data Distribution Service applications can use RTI Persistence Service from any platform. The Persistence Service daemon itself is supported on the set of platforms below:

#### Supported Database Systems

- TimesTen v6.0.4
- MySQL v5.0

#### Supported Operating Systems

- Linux 2.6 (32 and 64 bit)
- Solaris 2.10 (32 and 64 bit)
- Windows 2000/Windows XP Professional (32 bit)

#### Supported Architectures

- UltraSPARC
- X86

Note: RTI continually adds support for new database systems, operating systems and architectures. Also, not all combinations of operating systems and architectures are available. Please contact RTI for additional availability and supported combinations.

### About RTI

Real-Time Innovations (RTI) works in partnership with its customers to develop and integrate the world's most demanding real-time applications. RTI takes the risk out of distributed application development and system integration by providing deep expertise in real-time communications coupled with the highest performance messaging middleware. The company's software and services have been leveraged in a broad range of industries including defense, intelligence, simulation, industrial control, transportation, finance, medical and communications. Founded in 1991, RTI is privately held and headquartered in Sunnyvale, California. For more information, please visit [www.rti.com](http://www.rti.com).