



PubSub+ lets applications retrieve information weeks or months after it was initially sent.

PubSub+ is the first event broker to offer message replay not just in context of streaming, but with message exchange patterns like publish/subscribe, queueing, and request/reply.

Now you can combine the advantages of long-term message storage with the flexibility of using open protocols and a variety of message exchange patterns.

PubSub+ message replay functionality will help you improve the stability of your infrastructure, bring new applications to market faster, and eliminate the cost and complexity associated with third-party replay solutions.

Why is Message Replay important?

The potential costs to a business for losing data can be enormous, both to their bottom line and to their reputation. Replay functionality lets you initialize new applications with previously published data. It also protects applications by giving them the ability to easily recover from application issues such as misconfigured subscriptions, application bugs, or corrupt databases by redelivering data to the application.

Common Use Cases

- Giving new applications access to data that has already been published.
- Using production data in a dev or test environment to test new applications with real-world data.
- Repopulating corrupted databases.
- Fixing applications with misconfigured subscriptions by replaying data after subscriptions are fixed.

How Message Replay Works

With replay functionality enabled, PubSub+ event brokers store persistent messages in a replay log so applications can request and replay messages based on either queue name or topics.

When message replay is initiated, messages in the replay log starting from the requested start time are delivered to the client.

Messages can be replayed from a specific date and time, or as far back as the replay log extends. The replay stream eventually joins the stream of live messages without requiring any action from the consuming application.

Key Features

- Message replay can be initiated administratively for applications using any protocol or API, including AMQP, MQTT, REST/WebHook.
- Replay can be initiated on a per-client basis by the application itself via Solace APIs.
- Messages can be replayed to topics and queues used by JMS1.1 and JMS2.0 clients.
- Messages are replayed in the order they were originally received by the broker, with order preserved even across multiple topics.
- Wildcard subscriptions can be used to select which messages should be replayed while still maintaining the original publish order across these many topics.