

# Recovery Procedures

*This document contains information on Market Data recovery methods in case of network outage or late join. Applies to RTS Market Data Platform 5.1.0*

## Recovery Methods

There are several methods of recovering missed messages or synchronizing client systems to the latest market state. These methods have their own benefits and drawbacks. Clients can choose what to use in each particular outage type and data channel. Implementation of some recovery methods may not be required depending on client's needs.

## Recommended Recovery Strategies

In case of any problems, client systems should **synchronize to latest market state** as soon as possible. If complete **market history** is required, a message recovery could be initiated after successful synchronization.

The following table shows different aspects of Per-message and Full snapshot recovery methods:

Aspect		Per-message	Full snapshot
<b>Data flows</b>	Recovery data	Full historic replay	Only current state
	Completeness	All lost messages	None of lost messages
<b>Recovery delays</b>	Late joiners	– High	+ Constantly significant
	Minor outages	+ Low	– Constantly significant
	Major outages	– Significant	+ Constantly significant
<b>Bandwidth use</b>	Minor outages	+ Low	– High for large markets
	Major outages	– Significant	+ Same as for minor outages
	Late joiners	– High	+ Same as for minor outages

The recovering process applies to affected channels only. If instrument-level sequencing is implemented, client system could determine which particular instruments are affected by network outage. Unaffected channels & instruments can continue normal operations.

## Late Join

In case of late join the client system shall use one of the snapshot recovery methods to receive the current state of the market and start interpreting real-time updates with minimum delay. If the system requires a history of events in the market it may request it at any time before the sequence reset, preferably during non-trading hours or the hours of smaller market event rates. Per-message recovery is used to restore the history of events.

Recovery procedure should be repeated for each **market segment** independently:

1. Subscribe to **Incremental Refresh** channel.
2. Subscribe to **Market Snapshot** channel and recover market data for all instruments.
3. **Unsubscribe** from Market Snapshot channel when all instruments are recovered.

## Minor Outage

In case of a minor outage, when a small number of Incremental Refresh messages is missing, it is most efficient to use per-message recovery. However, snapshot recovery can also be used to reduce the implementation cost. Since per-message recovery delays are not guaranteed, the minimum recovery time may be achieved by combining per-message recovery with snapshot recovery, whichever fills the gap first.

## Major Outage

The recovery strategy for major outages shall be the same as in case of late join. However, if there is a need to restore the history of events during the outage, the sequence number of the last message processed before it shall be remembered. When the client system requests the history of messages it shall use this number as the start of the message range requested.

## TCP-based Recovery Methods

### On-demand Market Snapshot Recovery

Snapshot messages, also known as 'Full refresh', are used to synchronize client systems to the actual market state in case of any **outages**, **network problems** or **late join**.

This recovery method allows fastest possible synchronization in case of major outages.

Each snapshot message includes sequence number of the latest incremental refresh message included into this snapshot. This sequence number should be used for synchronization of instrument state delivered in snapshot and real-time incremental refresh messages.

1. Prepare:
  - 1.1. Begin **queuing incremental data** (stay subscribed to incremental refresh channel).
  - 1.2. Identify which security is affected using **RptSeq** while analyzing the incremental data being queued or assume that all **instruments** in a channel are affected (if instrument-level sequencing is not in use).
  - 1.3. **Open TCP connection** with Market Recovery service and send appropriate **login** message.
2. Send **Market Data Request** message to request a **Market Snapshot** for affected securities.
3. Receive current market state:
  - 3.1. Process all **Market Data Snapshot** messages and recover instrument state (depending on subscription type) from a Snapshot.
  - 3.2. After sending last Snapshot message Market Data Platform will **Logout** and close TCP connection.
4. Apply queued Market Data:
  - 4.1. All queued incremental data for specific Security with MsgSeqNum less or equal to Snapshot's **LastMsgSeqNumProcessed** message should be discarded.
  - 4.2. All incremental refresh messages received after (as of LastMsgSeqNumProcessed field) corresponding Snapshot message should be applied to instrument data recovered from the snapshot message. This way instrument data will be synchronized with real-time Incremental Refresh flow. When all affected instrument on the channel will be recovered, client system should unsubscribe from Snapshot channels.

Single Market Data Snapshot message will be sent for each instrument:

Channel	Market Data Entry type	
Level 1	Quotes / Last Best Price	Current best bid & offer prices
	Trade	Last trade price & volume
	Market Fundamentals	Trade session high/low prices
Level 2	Price levels	Current market depth data
Level 3	Full order book	All currently active orders

## Per-message Recovery

Per-message recovery is used to retransmit messages in **lost UDP packets**. This approach suitable in case of **minor outages** only and it isn't recommended in case of major outages or late joins. This recovery method allows requesting missed data packets and applying Incremental Refresh messages in correct sequence.

This recovery method allows fastest possible synchronization in case of minor outages.

1. Prepare:
  - 1.1. Identify which messages (**MsgSeqNum**) are missing.
  - 1.2. Begin **queuing incremental data** (stay connected to UDP multicast group).
  - 1.3. **Open TCP connection** with Market Recovery service and send appropriate **login** message.
2. Send **Market Data Request** message to request resend of **range of Incremental Refresh messages**.
3. Receive missed Incremental Refresh messages:
  - 3.1. After sending last message from the requested range, Market Data Platform will **Logout** and close TCP connection.
4. Apply received Market Data
  - 4.1. All **received and queued** incremental data should be processed in **right order**, designated by their MsgSeqNum fields.

## UDP-based Recovery Methods

UDP does not guarantee reliability or ordering in the way that TCP does. UDP packets may arrive out of order, appear duplicated, or go missing without notice. Avoiding the overhead of checking whether every packet actually arrived to counterparty makes UDP faster and more efficient.

This section provides supplemental methods to the recovery process, however they are not guaranteed to completely recover missed messages or synchronize client systems to the latest state.

## Multiple Incremental Refresh UDP Channel

It is recommended that client systems process both the A and B incremental market data UDP channels. These redundant channels broadcast same messages.

If a message gap is detected in Incremental Refresh Message Channel A, client systems should try to find missing messages in Channel B. If missing message wasn't found in both channels in reasonable time, recovery procedure should be initiated. It should be assumed at this point that all books maintained in the client system may no longer have the correct, latest state.

## Continuous Market Snapshot Recovery

This approach is similar to “On-demand market snapshot recovery”.

Main differences are:

- The service is non-interactive: **all snapshots** are published at least every 2 minutes.
- **Unreliable UDP** protocol is used for snapshots broadcasting – some packets might disappear.

### Recovery Steps

1. Prepare:
  - 1.1. Begin **queuing incremental data** for affected instruments.
  - 1.2. Identify which security is affected (using **RptSeq**) or assume that all **instruments** in a channel are affected.
  - 1.3. **Subscribe** to corresponding Market Recovery UDP channel.
2. Receive Market Data:
  - 2.1. Decode **Market Data Snapshot** messages received.
  - 2.2. Message can be ignored if its LastMsgSeqNumProcessed equals or less than last known MsgSeqNum for the instrument (normal operation).
3. Apply received Market Data:
  - 3.1. Update the book when Market Data Snapshot for affected instrument is received. The queued incremental data should then be applied as necessary.

### Instrument-level Sequencing

Market Data Incremental Refresh messages contain instrument sequence numbers, in addition to message sequence numbers (tag 34, MsgSeqNum). Every repeating group in a market data entry contains an incrementing sequence number (tag 83, RptSeq) that is associated with the instrument for which data is present in the block.

Instrument level sequencing can be used to identify which instruments you have no missed messages for, and filter recovery messages accordingly.

Client systems can keep track of the instrument sequence numbers (tags 83-RptSeq and 34-MsgSeqNum) for every instrument. This can be used to tell damaged and correct order books apart when some messages were lost:

- If there is no gap in instrument sequence number, the data can be used immediately, and it also indicates that the book for this instrument still has a correct, actual state.
- If there is a gap, it indicates that some data was missed for the instrument when message loss occurred.

Message sequence numbers can be used to determine if particular Market Snapshot should be applied (with help of field LastMsgSeqNumProcessed).