TASK 11 - Understand TP-RT Subscription and Replay Concepts

TASK 10 - Event Logging

TASK 12 - Timestamping Messages

NECESSARY KNOWLEDGE

n/a

I. AIM

The purpose of this task is to understand the concepts underlying the tickerplant->realtime database subcription and replay and to update the sequence number when new messages are logged.

II. INFORMATION

Recall that TP logs the data and RT holds the data in-memory. RT also saves down the data to disk at EOD, so it can be queried historically.

For this task we need to understand: TP<->RT interplay, what and how does TP log, what happens on RT subscription (or any client subscription)?

TP logs the incoming messages as **event streams** which it is publishing downstream, such that these events can be replayed by any subscribing client if necessary. Events are simply mixed lists of the form ***(callback function;table;data)***, e.g. (`upd;`trade;,(09:00:00.000;`BARC.L;100.1;10000)). TP also keeps track of the number of events logged, we call this the **sequence number**. On every logged event, we increase the sequence number by one.

On RT subscription, TP will return log name as well as the sequence number to RT; while **at the same time publishing incoming events downstream**. RT will start replaying the log up to the sequence number it received. **Note**: we use sequence number as the replay cutoff point, such that we won't end up with duplicate data, as the TP will log events while also sending them to RT during the time of replay.

III. INSTRUCTIONS

Open tp.q in your text editor and find the line:

```
upd:{[t;d]
```

Extend the function so that when new messages are logged the sequence number is increased.

IV. TESTING

To test your code in one session start the tickerplant process:

```
q tp.q -p 5000 -tp_path /tmp
```

In a second session start the feedhandler:

```
q fh.q -p 4000 -tp localhost:5000
```

In a third session start the simulator:

```
q simu.q -fh localhost:4000 -data data/msgs
```

In the simulator session type:

```
do[10;pub[]]
```

Then return to the tickerplant session and type:

```
get L
```

The the function should return something similar to:

```
`upd `trade (,`BARC.L;,289.2674;,1274743)                                    ..
`upd `trade (,`BARC.L;,289.6213;,352043)                                     ..
`upd `quote (,`VOD.L;,187f;,193f;,2500;,1800)                                ..
`upd `quote (`BARC.L`BARC.L`BARC.L`BARC.L;283 283 283 283f;300 300 300 300f;3..
`upd `quote (,`BARC.L;,285f;,300f;,6144;,76116)                              ..
`upd `quote (,`BARC.L;,285f;,300f;,6144;,105127)                             ..
`upd `quote (,`BARC.L;,285f;,300f;,6144;,191232)                             ..
`upd `quote (,`BARC.L;,290f;,300f;,4800;,191232)                             ..
`upd `trade (,`BASFn.DE;,67.5;,32529)                                        ..
`upd `trade (`BASFn.DE`BASFn.DE`BASFn.DE;67.5 67.5 67.5;100 150 500)         ..
`upd `quote (,`BASFn.DE;,67.51;,67.59;,201;,201)                             ..
`upd `quote (,`BASFn.DE;,67.51;,67.58;,201;,1)                               ..
```

Then type:

```
SEQ~-11!(-2,L)
```

1b should be returned.

TASK 10 - Event Logging

TASK 12 - Timestamping Messages