

Search all content

KX Community > Discussion Forums > q-bies > Re: What is the execution flow of this file

Options

# What is the execution flow of this file

Go to solution



mamsds

New Contributor II



a week ago

I am studying the official example code from here: <https://github.com/kxcontrib/websocket/blob/master/AppendixB/pubsub.q> . After some research, I can get the publisher/subscriber model running and I can understand the basic idea on how the feedhandler part (i.e., this file: <https://github.com/kxcontrib/websocket/blob/master/AppendixB/fh.q>) works. However, I am still very confusing on how the publisher part (i.e., this file: <https://github.com/kxcontrib/websocket/blob/master/AppendixB/pubsub.q>) works.

I understand that when the code is initially executed, it listens on port 5001 and it defines a few tables (trade, quote, subs) and functions (loadPage, filterSyms, getSyms, getQuotes, getTrades, sub, pub). Does it mean that the interpreter directly jumps to the end of the file, that is, these two lines:

```
1 | .z.ts:{pub each til count subs};
2 | \t 1000
```

and starts something like a "main loop" in winForm development?

Suppose my guess is right, the loop keeps calling the function *pub* until there aren't any new rows in the table *subs*. But how does *subs* get any new rows in the first place?



2 Kudos

Reply

## 1 ACCEPTED SOLUTION



rocuinneagain KX

Contributor

In response to mamsds



5m ago

Yes your understanding is correct for those 4 items.

These tutorial videos may be useful:

- Introduction to kdb+ and q #13: Interprocess Communication (\*)
- Introduction to kdb+ and q #14: Callbacks (\*)



0 Kudos

Reply

[All forum topics](#)   [Next Topic >](#)

5 REPLIES

Sort:   Oldest to Newest



**rocuinneagain** KX  
Contributor



Monday - last edited Monday

`z.ts` is the timer function which is evaluated on intervals of the timer variable set by system command `\t`

In your example 'pub' is called for each subscriber in 'subs' once per second.

To start subs has count 0 but when a new subscriber connects and subscribes they will be added.

How the subscribers are added to that table in that demo is:

1. websockets.html calls connect() on load of the page ([code](#))
2. websockets.js holds the connect() definition which shows it sends loadPage[] to the q process ([code](#))
3. pubsub.q holds the definition of loadPage which shows that it calls 'sub' ([code](#))
4. 'sub' then adds your subscriber to 'subs'



timer



2 Kudos

Accept as Solution

Reply



**mamsds**  
New Contributor II



In response to **rocuinneagain**



Tuesday - last edited Tuesday

Thanks @rocuinneagain for taking a look at my lengthy question! Let's say we examine the flow this way:

1. information is generated by feedhandler fh.q (<https://github.com/kxcontrib/websocket/blob/master/AppendixB/fh.q>)
2. information is passed to tickerplant subpub.q (<https://github.com/kxcontrib/websocket/blob/master/AppendixB/pubsub.q>) (please correct me if this is not a tickerplant)
3. information is received by a JavaScript-based subscriber (<https://github.com/kxcontrib/websocket/blob/master/AppendixB/websockets.js>)

In this particular example, I understand that information is randomly generated by fh.q; and fh.q interact with pubsub.q via these two function calls (<https://github.com/kxcontrib/websocket/blob/ad2f0b268afaeelfc5f4dda2fc2467440c7e2f0c/AppendixB/fh.q#...>

My confusion starts here: in lines 18 and 19 of fh.q, `upd` is an alias to insert; `quote` and `trade` are table names, and the rest are parameters randomly generated--so which line in pubsub.q will be invoked when these two function calls are made (please correct me if the term "function call" is inaccurate)?



0 Kudos

Accept as Solution

Reply



Tuesday - last edited Tuesday



**rocuinneagain** KX

Contributor

In response to **mamsds**

TP = Tickerplant (Relays data to subscribers + recovers from logfile after crashes)

RDB = Realtime Database (Stores data for query)

RTE = Realtime engine (Performs streaming calculations and stores caches or publishes results)

(Any process can be customised)

1. Yes

2. It is acting more like a mixture between a TP,RDB,RTE

- a) It does not store a logfile to recover in case of a crash (tp-logfile - a TP normally does this)
- b) It stores data indefinitely instead of acting only as a relay. (Unlike a TP, more like and RDB, although an RDB will clear once every 24hrs)
- c) It does not relay data untouched instead only specific data is forwarded (similar to an RTE)

getSyms - sends like of unique symbols across tables

getQuotes - sends last row by sym from quote table

getTrades - sends last row by sym from trade table

3. Yes

The execution flow is:

1. FH sends messages to PubSub (lines 18/19) every 100ms
2. The messages arrive to PubSub and z.pg evaluates them. This mean upd/insert (pubsub.q line 8 will save the incoming data to quote/trade. PubSub now has some data cached.
3. The next time the PubSub timer (.z.ts) is triggered (every second) the 'pub' function will trigger and send data to subscriptions.

This code is a basic demo so it may have some holes in it's logic (like never clearing data in PubSub so eventually memory will run out)



2 Kudos

Accept as Solution

Reply



Thursday - last edited 44 seconds ago



**mamsds**

New Contributor II

In response to **rocuinneagain**

Hi @rocuinneagain, thanks for the detailed reply and some informative clarification!

But I still have some confusion...the crux is this sentence you wrote

GroupsWelcome and NewsDiscussion ForumsHelp

2. The messages arrive to PubSub and .z.pg evaluates them. This mean upd/insert (pubsub.q line 8 will save the incoming data to quote/tr...

I understand that .z.pg is a function under the .z namespace, my questions are:

- Can I say that it is like a callback function which will be called by q when some events (i.e. synchronous request in this case) happen?
- In the caller side (<https://github.com/kxcontrib/websocket/blob/ad2f0b268afaeelfc5f4dda2fc2467440c7e2f0c/AppendixB/fh.q#...>), since we called the handler this way: `h(upd;quote;(n#.z.N;s;getbid'[s];getask'[s]))`;, does it mean that .z.pg will evaluate the parameters, i.e., ``upd, `quote` and `(n#.z.N;s;getbid'[s];getask'[s])` and .z.pg will interpret these parameters as: "execute the function upd and pass two parameters (i.e., ``quote` and `(n#.z.N;s;getbid'[s];getask'[s])`) to it"?
- Regarding this statement: "*This code is a basic demo so it may have some holes in it's logic (like never clearing data in PubSub so eventually memory will run out)*", do you mean that if feedhandler constantly inserts new rows by calling `h(upd;quote;(n#.z.N;s;getbid'[s];getask'[s]))`;, tables ``quote` and ``trade` will grow larger and larger and the memory will be filled by these two tables?
- Regarding this function (<https://github.com/kxcontrib/websocket/blob/ad2f0b268afaeelfc5f4dda2fc2467440c7e2f0c/AppendixB/pubsub.q#...>), due to brevity of kdb/q, I am still decoding what exactly it is doing. But after reading you last reply, my vague understanding is that subscribers registered their handlers in the table ``subs`. So the pub function is essentially enumerating the ``subs` table and call all subscribers' handlers one by one. This series of function calls is what we say to "publish data" to subscribers. Is this understanding basically correct?

Thanks!

0 Kudos

Accept as Solution

Reply

5m ago

**rocuinneagain**  
Contributor  
In response to **mamsds**

☒

Yes your understanding is correct for those 4 items.

These tutorial videos may be useful:

- [Introduction to kdb+ and q #13: Interprocess Communication \(\\*\)](#)
- [Introduction to kdb+ and q #14: Callbacks \(\\*\)](#)

[View solution in original post](#)

0 Kudos

Reply

Reply to the topic...

Post Reply

Related Content

c.so file for arm32

in KX Technology 3 hours ago

Accessing JIRA APIs through KDB



What is the execution flow of this file

in New kdb+/q users question forum a week ago

Cannot write to handle N. OS reports: Bad file descriptor

in New kdb+/q users question forum a week ago

Partitioning Tables Intraday by Custom Fields?

in KX Technology 2 weeks ago



Main Office Contacts

EMEA

Tel: +44 (0)28 3025 2242

AMERICAS

Tel: +1 (212) 447 6700

APAC

Tel: +61 (0)2 9236 5700

Useful Information

Technical Support Services Careers

Resources

Resource Library Use Cases Blog Events Training

Popular Links

Support.kx.com Code.kx KxSystems.Github.io ondemand Serverless Investor Relations

Follow Us

KX. All Rights Reserved.  
  
KX and kdb+ are registered trademarks of KX Systems, Inc., a subsidiary of FD Technologies plc.

Privacy Policy  
  
Terms of Use  
  
Export Statement  
  
Processing Notice