

NAME

bars – Build minute bars from Level I IQFeed stream data, storing results in Redis.

SYNOPSIS

bars [-h] [-I *iqfeedhost*] [-F *iqfeedport*] [-p *price*] [-s *symbol*] [-v *vol*] [-t *time*] [-H *redishost*] [-P *redisport*]

DESCRIPTION

The bars client for IQFeed parses Level I streaming data and builds minute bars for a user-specified set of symbols. A minute bar is a data structure that contains the first recorded trade price (open), the last recorded trade price (close), the highest trade price, the lowest trade price and total aggregate trade volume, for a one minute interval. Bars are stored in Redis approximately every minute using the key defined by ".BARS" appended to the symbol name. Other client applications (like R, Python, Octave, Matlab, etc.) can consult Redis at anytime to look up the day's minute bar data for a given symbol.

Minute bars for each symbol are stored in a binary array consisting of 6 columns by 1440 rows. The rows represent every daily minute interval from midnight to 23:59. The columns represent the open, high, low, close, and total volume bar fields, plus an extra column used internally for handling out of order ticks. Bars that have not yet been assigned data are initialized with NAN. Note that past bars may be updated after the corresponding minute interval has passed since ticks may arrive late.

Only quote and update events ('P' and 'Q' IQ Feed message types) that indicate trades are used to build the bars. The program also stores the last updated bar row index (starting at index 0) in the Redis key .barindex. The last update of market time is stored in the key .markettime. The most recent market quote is stored in text format for each symbol in Redis using the symbol name as the key.

The bars program parses the /etc/iqfeed.symbols file, if it exists, on start up. Symbols to monitor should appear in this file, one symbol per line.

Users may also send control signals to the bars program while running to add or remove watched symbols. The control port defaults to 7777, but may be set as a user option. See the examples below.

OPTIONS

-p *NUMBER*

Set the IQ Feed message field number corresponding to last trade price.

-s *NUMBER*

Set the IQ Feed message field number corresponding to symbol.

-t *NUMBER*

Set the IQ Feed message field number corresponding to time stamp.

-v *NUMBER*

Set the IQ Feed message field number corresponding to volume.

-u *NUMBER*

Bars updates Redis at least *NUMBER* seconds past each minute. The default value is 5 seconds, and must be less than 30 seconds.

-H *HOST*

Set the Redis host name or IP address.

-P *NUMBER*

Set the Redis port number.

-F *NUMBER*

Set the IQ Feed port number.

-I *HOST*

Set the IQ Feed host name or IP address.

NOTES

The bars program divides each day into 1,440 bars. The first bar records data with market time stamps lying between midnight (inclusive) and 12:01 (exclusive). The second bar records market data with time stamps between 12:01 (inclusive), and 12:02 (exclusive), and so on.

Redis keys associated with minute bar data are set to expire every night at midnight. However, in order to surely avoid confusion, do not use minute bars with indices greater than the last updated index in the key `.barindex`.

EXAMPLES

Connect to IQ Feed on the remote 'Windows' host on port 4009:

```
bars -I Windows -F 4009
```

Connect to IQ Feed on the local machine on the standard IQ Feed port:

```
bars -I localhost -F 5009
```

Instruct the feed to emit events for the equity symbol MSFT:

```
echo wMSFT | nc localhost 7777
```

The Redis key 'MSFT.BARS' will now contain the day's minute bar data for the symbol MSFT in binary array form. The Redis key 'MSFT' now contains the last raw market tick data. The Redis key ".barindex" contains the last bar row index that was updated.

Instruct the feed to stop watching MSFT:

```
echo rMSFT | nc localhost 7777
```

Pull the bars into an R matrix object with, for example:

```
require(rredis)
redisConnect()
index <- redisGet(".barindex") + 1 # Shows us the current bar index
msft <- matrix(readBin(redisGet('MSFT.BARS', raw=TRUE),
  what='double',n=1440*6),byrow=TRUE,nrow=1440)[,1:5]
colnames(msft) <- c('open','high','low','close','totalVolume')
```

AUTHOR

Written by B. Wayne Lewis, illposed.net.

COPYRIGHT

None.