EXAMPLE: How to build a dYdX Orderbook

Author: Lawrence Chiu (lawrence@dydx.exchange)

Twitter: https://twitter.com/LawrenceChiu14

Date: 4/8/2022 (updated 4/11/2022)

Tested on: Ubuntu Server 20.04.4 LTS

We will build the dYdX orderbook with 3 programs:

1) dydxob.py (the orderbook builder).

2) dydxtrades (the last-trade monitor), and

3) dydxob2.py (the orderbook displayer)

All programs are available at https://github.com/chiwalfrm/dydxexamples

PART 1: dydxob.py

1. First, create a 1GB ramdisk and mount it as /mnt/ramdisk/ as this program will write a LOT of files to it continuously. Set the permissions mask to 777.

```
$ sudo mount -t tmpfs -o rw,size=1G tmpfs /mnt/ramdisk

$ sudo chmod 777 /mnt/ramdisk
```

2. Next, start the dydxob.py program. Invoke the program with the <MARKET> parameter (if omitted, it defaults to 'BTC-USD'). You can get a list of markets from https://api.dydx.exchange/v3/markets Markets with status = 'POST_ONLY' are not open for trading, and this program will not do anything. The -u option tells python not to buffer.

```
-u Force the stdout and stderr streams to be unbuffered.  This option
has no effect on the stdin stream.
```

```
$ nohup python3 -u dydxob.py BTC-USD > /mnt/ramdisk/dydxobBTC-USD.stdout 2>&1 &
```

3. This creates and continuously updates 'price' files in the directories <MARKET>/asks/ and <MARKET>/bids/. These price files each contain four elements: the 'offset' of the price, the 'size' at that price, and the 'date' (yyyy-mm-dd) and 'time' (hh:mm:ss) of the last update. The program logs to and automatically rotates the logfiles /mnt/ramdisk/dydxob<MARKET>.log so that each <MARKET> uses less than 5 MB.

```
vmware@ubuntu20041a:~$ cat /mnt/ramdisk/BTC-USD/asks/42863
12240989790 0 2022-04-10 12:17:37
vmware@ubuntu20041a:~$
```

PART 2: dydxtrades.py

1. Start the dydxtrades.py program. Invoke the program with the <MARKET> parameter (if omitted, it defaults to ‘BTC-USD’).

```
$ nohup python3 -u dydxtrades.py BTC-USD > /mnt/ramdisk/dydxtradesBTC-USD.stdout 2>&1 &
```

2. This creates and continuously updates the file <MARKET>/lasttrade with the ‘date/timestamp’, ‘price’ and ‘side’ (BUY or SELL) of the last trade. The program logs to and automatically rotates the logfiles /mnt/ramdisk/dydxtrades<MARKET>.log so that each <MARKET> uses less than 5 MB.

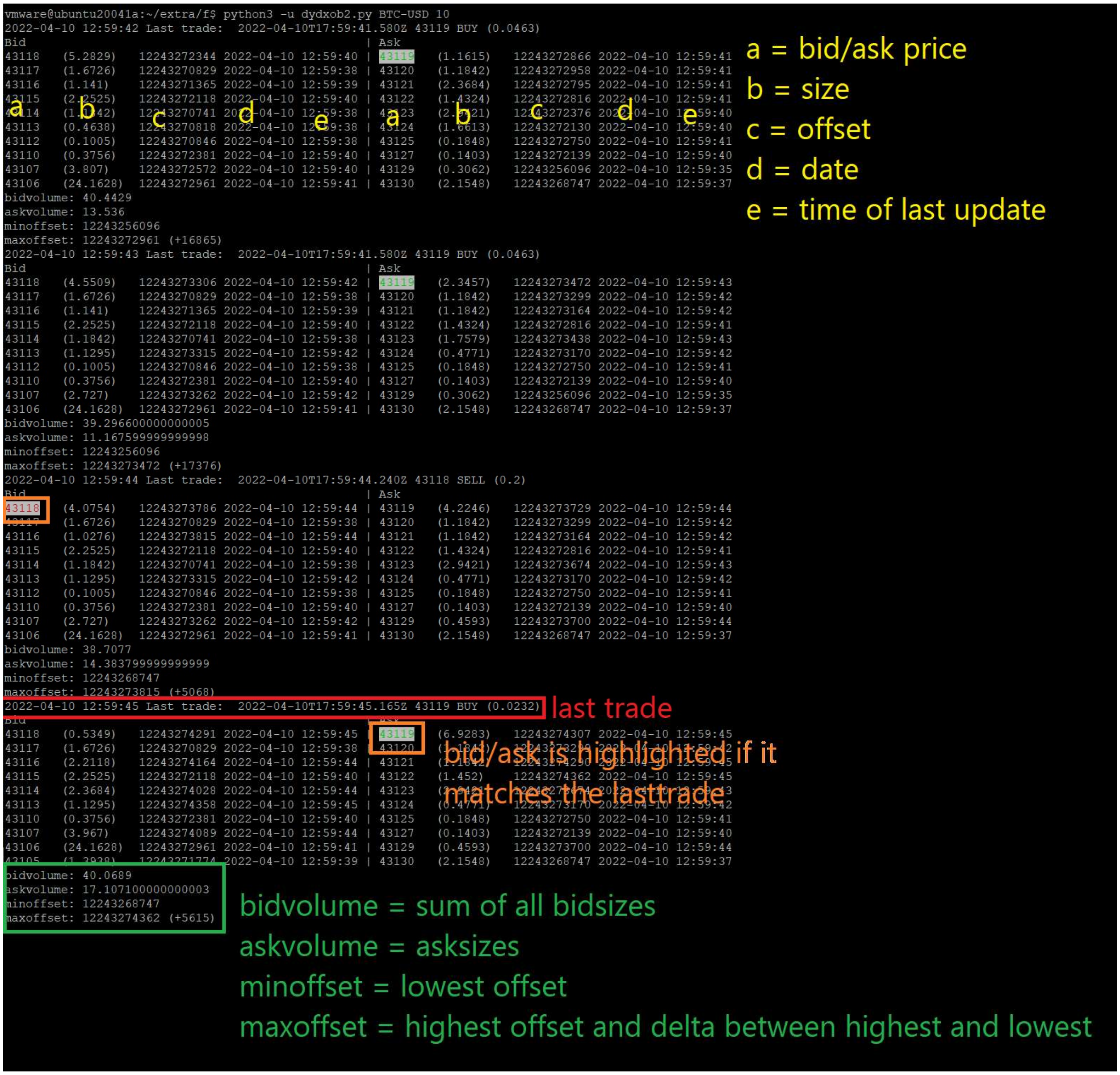
```
vmware@ubuntu20041a:~/extra/f$ cat /mnt/ramdisk/BTC-USD/lasttrade
2022-04-10T17:51:15.687Z 43056 SELL (0.0229)
vmware@ubuntu20041a:~/extra/f$
```

PART 3: dydxob2.py

1. Finally, start the dydxob2.py program. Invoke the program with the <MARKET> and <DEPTH> parameters. <MARKET> must be first parameter, and <DEPTH> must be second. If <DEPTH> is omitted, it defaults to '10'. If <MARKET> and <DEPTH> are both omitted, it defaults to 'BTC-USD' and '10'. This requires a 103-column terminal to look good.

```
$ python3 -u dydxob2.py BTC-USD 10
```

2. This will display and continuously update the order book on your screen.



3. (Optional) There are a few optional parameters to customize the display to your preference.
   a. compact – Removes 'offset' values from the display (67-column terminal required)
   b. ultracompact – Removes 'offset' and year part of the 'date' values (39-column terminal required)
   c. noansi – Removes colored values (this must be the last parameter)

```
vmware@ubuntu20041a:~/extra/f$ python3 -u dydxob2.py BTC-USD 10 compact
2022-04-10 13:28:18 2022-04-10T18:28:18.381Z 43206 SELL (0.2)
Bid                                      | Ask
43206   (4.6253)  04-10 13:28:18 | 43207   (1.5398)  04-10 13:28:18
43205   (1.7342)  04-10 13:28:16 | 43209   (1.1842)  04-10 13:28:17
43204   (1.1842)  04-10 13:28:16 | 43210   (3.2944)  04-10 13:28:18
43203   (2.4384)  04-10 13:28:18 | 43211   (1.4844)  04-10 13:28:17
43202   (1.16)    04-10 13:28:16 | 43212   (0.2636)  04-10 13:28:17
43201   (1.0909)  04-10 13:28:17 | 43213   (0.2179)  04-10 13:28:17
43200   (2.2496)  04-10 13:28:16 | 43214   (0.7333)  04-10 13:28:17
43199   (3.6899)  04-10 13:28:17 | 43216   (0.6727)  04-10 13:28:18
43196   (0.7406)  04-10 13:28:17 | 43217   (5.3617)  04-10 13:28:18
43195   (0.6412)  04-10 13:28:15 | 43218   (0.3056)  04-10 13:28:16
bidvolume: 19.554300000000005
askvolume: 15.057600000000003
minoffset: 12244538054
maxoffset: 12244540983 (+2929)
```

```
vmware@ubuntu20041a:~/extra/f$ python3 -u dydxob2.py BTC-USD 10 ultracompact
04-10T18:28:58.481Z 43197 BUY (0.0462)
Bid                | Ask
43196   (6.932)    | 43197   (3.9598)
43195   (2.9314)   | 43198   (1.1842)
43194   (3.4008)   | 43199   (1.1842)
43193   (1.1842)   | 43200   (0.7081)
43192   (0.1274)   | 43201   (0.55)
43191   (1.1842)   | 43207   (0.5206)
43190   (0.6459)   | 43208   (0.1389)
43187   (27.5577)  | 43210   (0.463)
43186   (2.7433)   | 43211   (0.3002)
43185   (29.2198)  | 43212   (0.6019)
bidvolume: 75.9267
askvolume: 9.610899999999999
minoffset: 12244551304
maxoffset: 12244560171 (+8867)
```

```
vmware@ubuntu20041a:~/extra/f$ python3 -u dydxob2.py BTC-USD 10 ultracompact noansi
04-10T18:29:23.831Z 43203 SELL (0.1)
Bid                | Ask
43203   (4.1436)   | 43204   (0.6842)
43202   (2.0784)   | 43205   (1.1311)
43201   (1.6726)   | 43206   (1.1842)
43200   (4.7994)   | 43207   (0.4528)
43199   (1.1842)   | 43210   (0.6158)
43198   (4.3342)   | 43211   (0.3002)
43195   (0.563)    | 43212   (3.6438)
43194   (2.3556)   | 43213   (0.4957)
43192   (1.1508)   | 43217   (1.7619)
43191   (0.278)    | 43218   (2.0317)
bidvolume: 22.559799999999996
askvolume: 12.301400000000001
minoffset: 12244558946
maxoffset: 12244572834 (+13888)
```

PART 4: (Optional) Launching multiple orderbooks

1. You can launch this program for multiple dYdX markets with this simple command.

```
for s1 in 1INCH-USD AAVE-USD  ADA-USD ALGO-USD ATOM-USD AVAX-USD BCH-USD BTC-USD COMP-USD \
       CRV-USD DOGE-USD DOT-USD ENJ-USD EOS-USD ETC-USD ETH-USD FIL-USD LINK-USD LTC-USD \
       MATIC-USD MKR-USD SNX-USD SOL-USD SUSHI-USD UMA-USD UNI-USD XLM-USD XMR-USD YFI-USD \
       ZEC-USD ZRX-USD
do
       nohup python3 -u dydxob.py $s1 > /mnt/ramdisk/dydxob$s1.stdout 2>&1 &
       nohup python3 -u dydxtrades.py $s1 > /mnt/ramdisk/dydxob$s1.stdout 2>&1 &
       sleep 1
done
```

<END>