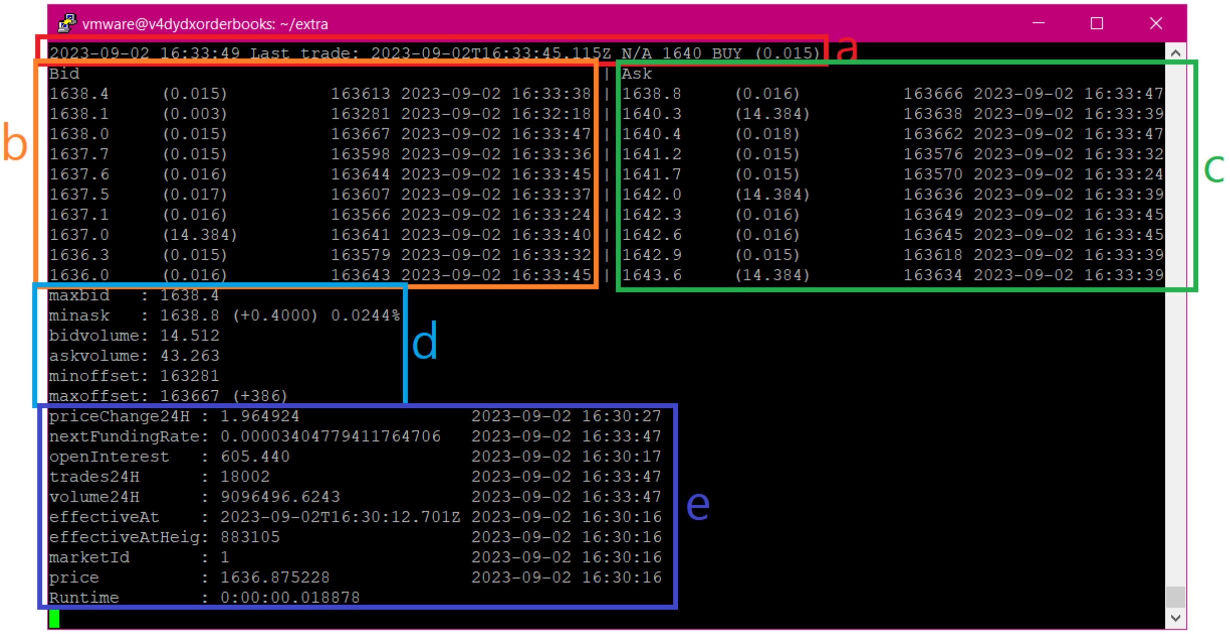
V4 dYdX Orderbook

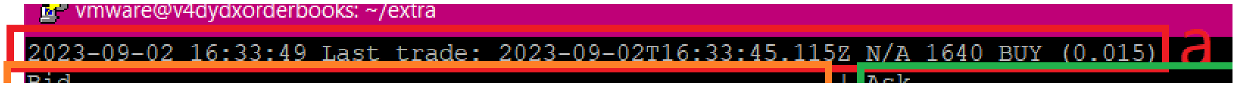
Author: Lawrence Chiu (lawrence@dydx.exchange)
Twitter: https://twitter.com/LawrenceChiu14
Date: 9/1/2023 (Updated 9/2/2023)
Tested on: Ubuntu Server 20.04.6 LTS

Let’s start off by looking at the end result first. This is the order book for ETH-USD. There are 5 areas to talk about, labeled ‘a’ to ‘e’ inclusive.



```
2023-09-02 16:33:49 Last trade: 2023-09-02T16:33:45.115Z N/A 1640 BUY (0.015)
Bid                                                     | Ask
1638.4      (0.015)           163613 2023-09-02 16:33:38 | 1638.8      (0.016)           163666 2023-09-02 16:33:47
1638.1      (0.003)           163281 2023-09-02 16:32:18 | 1640.3      (14.384)          163638 2023-09-02 16:33:39
1638.0      (0.015)           163667 2023-09-02 16:33:47 | 1640.4      (0.018)           163662 2023-09-02 16:33:47
1637.7      (0.015)           163598 2023-09-02 16:33:36 | 1641.2      (0.015)           163576 2023-09-02 16:33:32
1637.6      (0.016)           163644 2023-09-02 16:33:45 | 1641.7      (0.015)           163570 2023-09-02 16:33:24
1637.5      (0.017)           163607 2023-09-02 16:33:37 | 1642.0      (14.384)          163636 2023-09-02 16:33:39
1637.1      (0.016)           163566 2023-09-02 16:33:24 | 1642.3      (0.016)           163649 2023-09-02 16:33:45
1637.0      (14.384)          163641 2023-09-02 16:33:40 | 1642.6      (0.016)           163645 2023-09-02 16:33:45
1636.3      (0.015)           163579 2023-09-02 16:33:32 | 1642.9      (0.015)           163618 2023-09-02 16:33:39
1636.0      (0.016)           163643 2023-09-02 16:33:45 | 1643.6      (14.384)          163634 2023-09-02 16:33:39
maxbid   : 1638.4
minask   : 1638.8 (+0.4000) 0.0244%
bidvolume: 14.512
askvolume: 43.263
minoffset: 163281
maxoffset: 163667 (+386)
priceChange24H : 1.964924                   2023-09-02 16:30:27
nextFundingRate: 0.00003404779411764706     2023-09-02 16:33:47
openInterest   : 605.440                    2023-09-02 16:30:17
trades24H      : 18002                      2023-09-02 16:33:47
volume24H      : 9096496.6243               2023-09-02 16:33:47
effectiveAt    : 2023-09-02T16:30:12.701Z 2023-09-02 16:30:16
effectiveAtHeig: 883105                     2023-09-02 16:30:16
marketId       : 1                          2023-09-02 16:30:16
price          : 1636.875228                2023-09-02 16:30:16
Runtime        : 0:00:00.018878
```

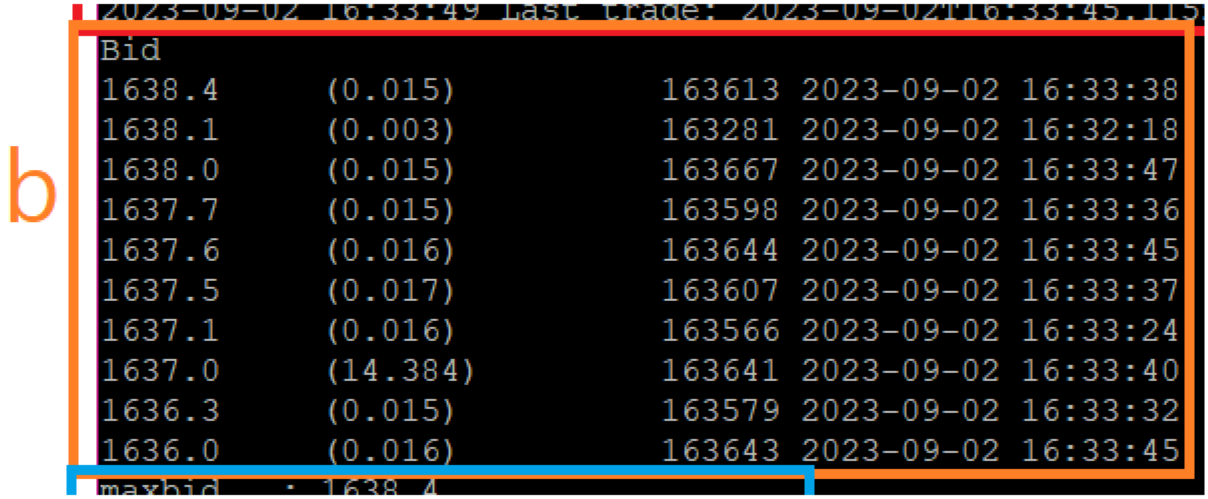
a. This is the last trade from the v4_trades websocket channel.



```
2023-09-02 16:33:49 Last trade: 2023-09-02T16:33:45.115Z N/A 1640 BUY (0.015)
```

From left to right, we got: 1) local server date and time, then 2) the trade’s date and time, 3) either ‘N/A’ here or the *tradecreatedatheight* value if it is a long-term trade, 4) the price, 5) side (BUY or SELL), and 6) size.

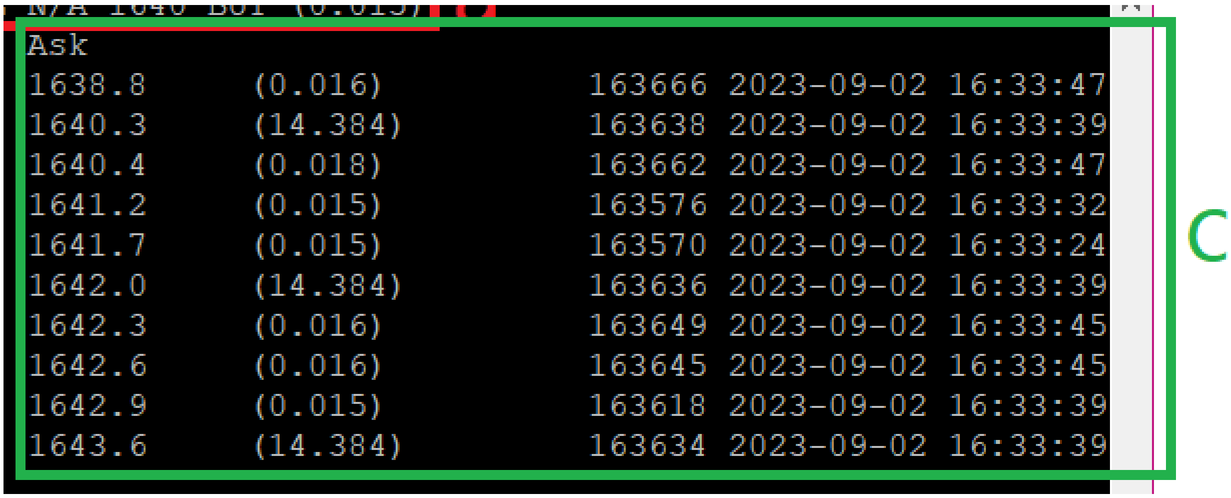
b. This is the list of bids from high price to low price.



```
2023-09-02 16:33:49 Last trade: 2023-09-02T16:33:45.115
Bid
1638.4      (0.015)         163613 2023-09-02 16:33:38
1638.1      (0.003)         163281 2023-09-02 16:32:18
1638.0      (0.015)         163667 2023-09-02 16:33:47
1637.7      (0.015)         163598 2023-09-02 16:33:36
1637.6      (0.016)         163644 2023-09-02 16:33:45
1637.5      (0.017)         163607 2023-09-02 16:33:37
1637.1      (0.016)         163566 2023-09-02 16:33:24
1637.0      (14.384)        163641 2023-09-02 16:33:40
1636.3      (0.015)         163579 2023-09-02 16:33:32
1636.0      (0.016)         163643 2023-09-02 16:33:45
maxbid    : 1638.4
```

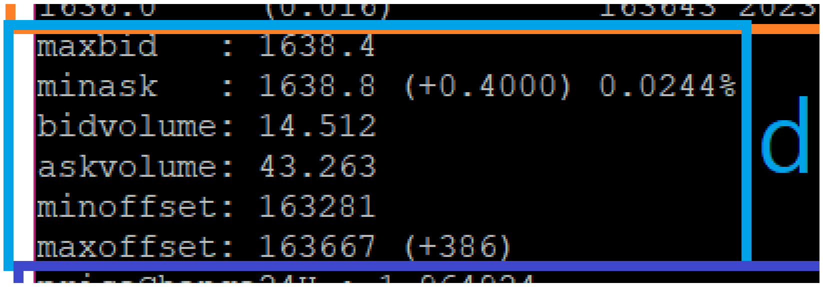
From left to right, we got: 1) price, 2) size, 3) the 'offset' (we'll talk about this below), and 4) the local server date and time (when the price was received on websocket).

c. Same thing as above except these are asks.



```
N/A 1640 B01 (0.015)
Ask
1638.8      (0.016)         163666 2023-09-02 16:33:47
1640.3      (14.384)        163638 2023-09-02 16:33:39
1640.4      (0.018)         163662 2023-09-02 16:33:47
1641.2      (0.015)         163576 2023-09-02 16:33:32
1641.7      (0.015)         163570 2023-09-02 16:33:24
1642.0      (14.384)        163636 2023-09-02 16:33:39
1642.3      (0.016)         163649 2023-09-02 16:33:45
1642.6      (0.016)         163645 2023-09-02 16:33:45
1642.9      (0.015)         163618 2023-09-02 16:33:39
1643.6      (14.384)        163634 2023-09-02 16:33:39
```

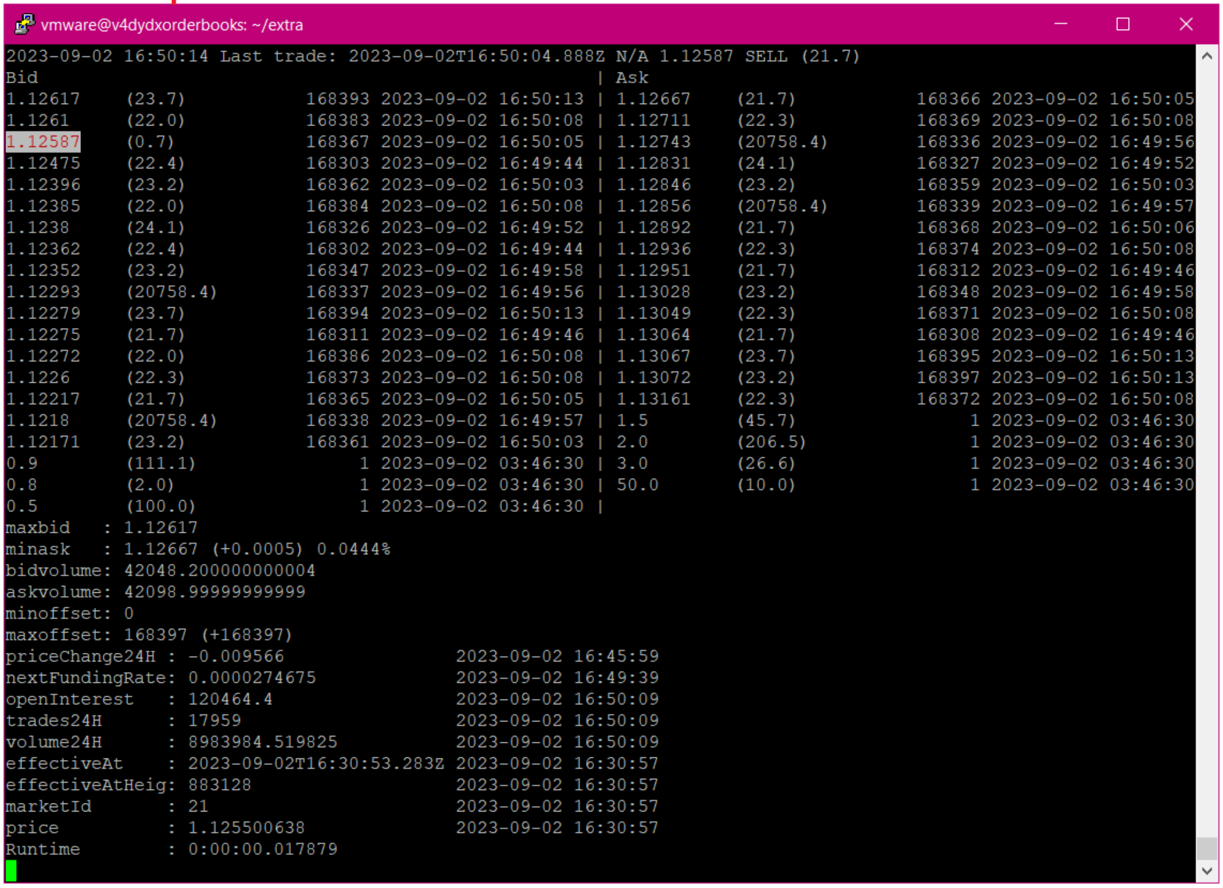
d. This shows some information (some calculated) from the bids/asks above.



```
1636.0      (0.016)         163643 2023
maxbid    : 1638.4
minask    : 1638.8  (+0.4000)  0.0244%
bidvolume: 14.512
askvolume: 43.263
minoffset: 163281
maxoffset: 163667  (+386)
priceChange24H : 1.064024
```

In particular, we see: 1) the highest bid, 2) the lowest ask along with a delta both in amount and percent, 3) the total bid volume, 4) the total ask volume, 5) the minimum offset, and 6) the maximum offset along with a delta. Note that these numbers will change if you choose a different depth level. In this example, depth = 10, so we see 10 bids and 10 asks. If you specify a depth that is larger than what's available, it'll show the entire side of the book. For example, below we see the orderbook for NEAR-USD with depth = 30. The order book only has 29 asks, so it has one fewer level than on the bid side.

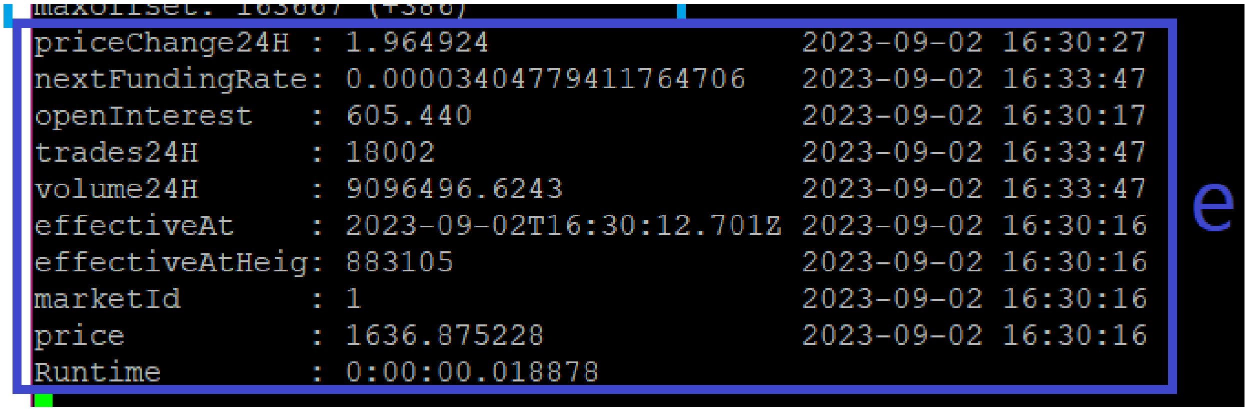
Example of an order book with fewer depth on one side vs the other



```
2023-09-02 16:50:14 Last trade: 2023-09-02T16:50:04.888Z N/A 1.12587 SELL (21.7)
Bid                                                    | Ask
1.12617    (23.7)           168393 2023-09-02 16:50:13 | 1.12667    (21.7)           168366 2023-09-02 16:50:05
1.1261     (22.0)           168383 2023-09-02 16:50:08 | 1.12711    (22.3)           168369 2023-09-02 16:50:08
1.12587    (0.7)            168367 2023-09-02 16:50:05 | 1.12743    (20758.4)        168336 2023-09-02 16:49:56
1.12475    (22.4)           168303 2023-09-02 16:49:44 | 1.12831    (24.1)           168327 2023-09-02 16:49:52
1.12396    (23.2)           168362 2023-09-02 16:50:03 | 1.12846    (23.2)           168359 2023-09-02 16:50:03
1.12385    (22.0)           168384 2023-09-02 16:50:08 | 1.12856    (20758.4)        168339 2023-09-02 16:49:57
1.1238     (24.1)           168326 2023-09-02 16:49:52 | 1.12892    (21.7)           168368 2023-09-02 16:50:06
1.12362    (22.4)           168302 2023-09-02 16:49:44 | 1.12936    (22.3)           168374 2023-09-02 16:50:08
1.12352    (23.2)           168347 2023-09-02 16:49:58 | 1.12951    (21.7)           168312 2023-09-02 16:49:46
1.12293    (20758.4)        168337 2023-09-02 16:49:56 | 1.13028    (23.2)           168348 2023-09-02 16:49:58
1.12279    (23.7)           168394 2023-09-02 16:50:13 | 1.13049    (22.3)           168371 2023-09-02 16:50:08
1.12275    (21.7)           168311 2023-09-02 16:49:46 | 1.13064    (21.7)           168308 2023-09-02 16:49:46
1.12272    (22.0)           168386 2023-09-02 16:50:08 | 1.13067    (23.7)           168395 2023-09-02 16:50:13
1.1226     (22.3)           168373 2023-09-02 16:50:08 | 1.13072    (23.2)           168397 2023-09-02 16:50:13
1.12217    (21.7)           168365 2023-09-02 16:50:05 | 1.13161    (22.3)           168372 2023-09-02 16:50:08
1.1218     (20758.4)        168338 2023-09-02 16:49:57 | 1.5        (45.7)                1 2023-09-02 03:46:30
1.12171    (23.2)           168361 2023-09-02 16:50:03 | 2.0        (206.5)               1 2023-09-02 03:46:30
0.9        (111.1)               1 2023-09-02 03:46:30 | 3.0        (26.6)                1 2023-09-02 03:46:30
0.8        (2.0)                 1 2023-09-02 03:46:30 | 50.0       (10.0)                1 2023-09-02 03:46:30
0.5        (100.0)               1 2023-09-02 03:46:30 |
maxbid   : 1.12617
minask   : 1.12667 (+0.0005) 0.0444%
bidvolume: 42048.200000000004
askvolume: 42098.99999999999
minoffset: 0
maxoffset: 168397 (+168397)
priceChange24H : -0.009566                2023-09-02 16:45:59
nextFundingRate: 0.0000274675             2023-09-02 16:49:39
openInterest   : 120464.4                 2023-09-02 16:50:09
trades24H      : 17959                    2023-09-02 16:50:09
volume24H      : 8983984.519825           2023-09-02 16:50:09
effectiveAt    : 2023-09-02T16:30:53.283Z 2023-09-02 16:30:57
effectiveAtHeig: 883128                   2023-09-02 16:30:57
marketId       : 21                       2023-09-02 16:30:57
price          : 1.125500638              2023-09-02 16:30:57
Runtime        : 0:00:00.017879
```

Let’s continue with our ETH-USD example.

e. This shows market parameters from the v4_markets websocket.



```
maxOffset: 163667 (+386)
priceChange24H : 1.964924                   2023-09-02 16:30:27
nextFundingRate: 0.000034047794117647O6     2023-09-02 16:33:47
openInterest   : 605.440                    2023-09-02 16:30:17
trades24H      : 18002                      2023-09-02 16:33:47
volume24H      : 9096496.6243               2023-09-02 16:33:47
effectiveAt    : 2023-09-02T16:30:12.701Z   2023-09-02 16:30:16
effectiveAtHeig: 883105                     2023-09-02 16:30:16
marketId       : 1                          2023-09-02 16:30:16
price          : 1636.875228                2023-09-02 16:30:16
Runtime        : 0:00:00.018878
```

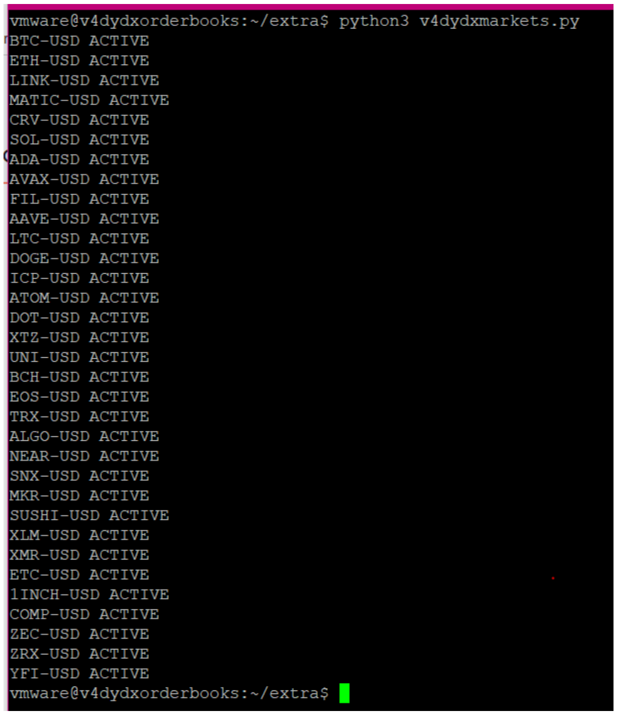
Don’t be concerned if you don’t see all of these parameters early on. The missing ones simply haven’t appeared on the websocket yet, but eventually you’ll see all of them. Finally, the last time (Runtime) tells you how long it took to render this data, in this case just 18ms.

Notes

1. Please read the V3 orderbook.pdf first at https://github.com/chiwalfrm/dydxexamples
2. V4 uses /mnt/ramdisk5 for the ramdisk (vs /mnt/ramdisk for V3). This allows you to run both V3 and V4 order books on the same computer.
3. The following are the V4 files with their V3 equivalents. They work exactly the same except on V4: “dydx-testnet-2” (Testnet Chain ID).

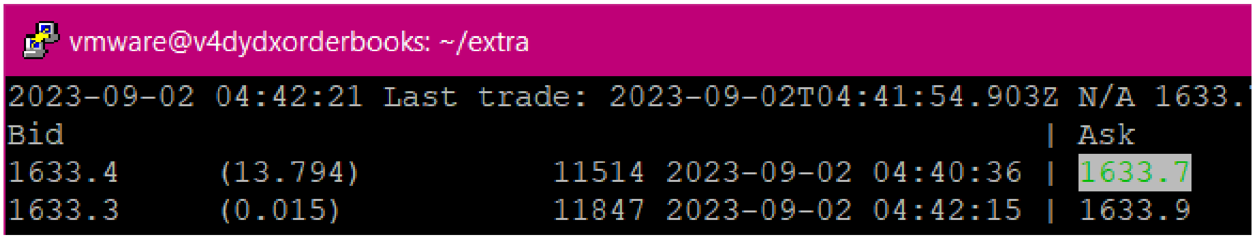
V4 program	**V3 program**
v4dydxob.py	dydxob.py
v4dydxtrades.py	dydxtrades.py
v4dydxv4markets.py	dydxv3markets.py
v4dydxob2.py	dydxob2.py

4. In addition, **v4dydxmarkets.py** can be used to list every market and their status (equivalent to **dydxmarkets.py** for V3). There are currently 33 total.



```
vmware@v4dydxorderbooks:~/extra$ python3 v4dydxmarkets.py
BTC-USD ACTIVE
ETH-USD ACTIVE
LINK-USD ACTIVE
MATIC-USD ACTIVE
CRV-USD ACTIVE
SOL-USD ACTIVE
ADA-USD ACTIVE
AVAX-USD ACTIVE
FIL-USD ACTIVE
AAVE-USD ACTIVE
LTC-USD ACTIVE
DOGE-USD ACTIVE
ICP-USD ACTIVE
ATOM-USD ACTIVE
DOT-USD ACTIVE
XTZ-USD ACTIVE
UNI-USD ACTIVE
BCH-USD ACTIVE
EOS-USD ACTIVE
TRX-USD ACTIVE
ALGO-USD ACTIVE
NEAR-USD ACTIVE
SNX-USD ACTIVE
MKR-USD ACTIVE
SUSHI-USD ACTIVE
XLM-USD ACTIVE
XMR-USD ACTIVE
ETC-USD ACTIVE
1INCH-USD ACTIVE
COMP-USD ACTIVE
ZEC-USD ACTIVE
ZRX-USD ACTIVE
YFI-USD ACTIVE
vmware@v4dydxorderbooks:~/extra$
```

5. The order book display a price in color if it matches the last trade. You can turn this off by specifying ‘noansi’ as the last parameter in the command line.

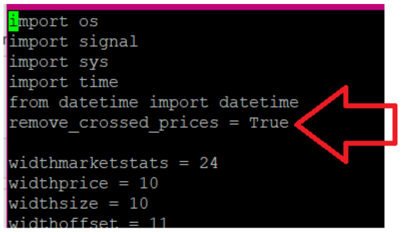


```
2023-09-02 04:42:21 Last trade: 2023-09-02T04:41:54.903Z N/A 1633.
Bid                                                       | Ask
1633.4      (13.794)          11514 2023-09-02 04:40:36 | 1633.7
1633.3      (0.015)           11847 2023-09-02 04:42:15 | 1633.9
```

In this example, the highlighted ask (1633.7) is the same price as the last trade.

About ‘offset’

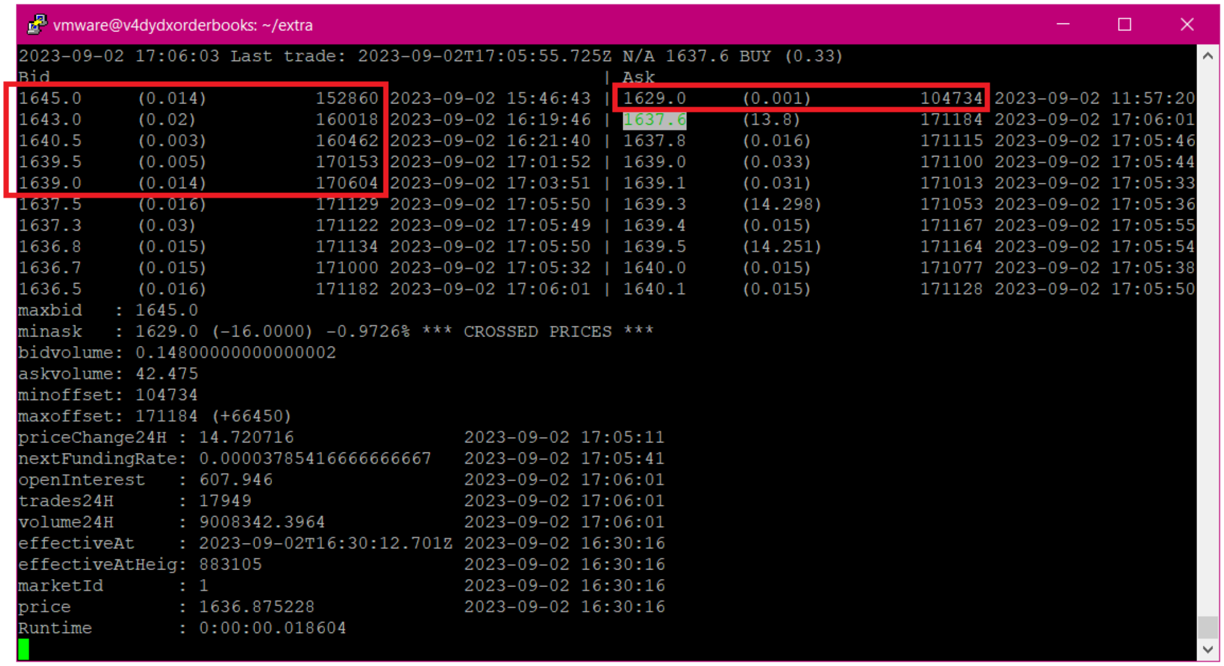
1. Because V4 does not have a centralized order book, there is no longer a global ‘offset’ parameter (like we had in V3) for order book prices. At any given time, the ‘correct’ order book is sitting in the mempool of the current block proposer. The block proposer changes every block, so that means a new canonical mempool and therefore a new canonical order book. Therefore, from time to time, you may see crossed prices from the indexer’s websocket. The way we deal with that in this implementation, is we assign every price, our own local offset (using the websocket’s *message_id*) and we assume that later prices have higher precedence. If you do not want to remove crossed prices, you can change **v4dydxob2.py**’s *remove_crossed_prices* to False



```
import os
import signal
import sys
import time
from datetime import datetime
remove_crossed_prices = True

widthmarketstats = 24
widthprice = 10
widthsize = 10
widthoffset = 11
```

2. Let’s take a look at an example of an order book with crossed prices (by setting the parameter above to False).



```
2023-09-02 17:06:03 Last trade: 2023-09-02T17:05:55.725Z N/A 1637.6 BUY (0.33)
Bid                                                     | Ask
1645.0     (0.014)          152860 2023-09-02 15:46:43 | 1629.0     (0.001)          104734 2023-09-02 11:57:20
1643.0     (0.02)           160018 2023-09-02 16:19:46 | 1637.6     (13.8)           171184 2023-09-02 17:06:01
1640.5     (0.003)          160462 2023-09-02 16:21:40 | 1637.8     (0.016)          171115 2023-09-02 17:05:46
1639.5     (0.005)          170153 2023-09-02 17:01:52 | 1639.0     (0.033)          171100 2023-09-02 17:05:44
1639.0     (0.014)          170604 2023-09-02 17:03:51 | 1639.1     (0.031)          171013 2023-09-02 17:05:33
1637.5     (0.016)          171129 2023-09-02 17:05:50 | 1639.3     (14.298)         171053 2023-09-02 17:05:36
1637.3     (0.03)           171122 2023-09-02 17:05:49 | 1639.4     (0.015)          171167 2023-09-02 17:05:55
1636.8     (0.015)          171134 2023-09-02 17:05:50 | 1639.5     (14.251)         171164 2023-09-02 17:05:54
1636.7     (0.015)          171000 2023-09-02 17:05:32 | 1640.0     (0.015)          171077 2023-09-02 17:05:38
1636.5     (0.016)          171182 2023-09-02 17:06:01 | 1640.1     (0.015)          171128 2023-09-02 17:05:50
maxbid   : 1645.0
minask   : 1629.0 (-16.0000) -0.9726% *** CROSSED PRICES ***
bidvolume: 0.14800000000000002
askvolume: 42.475
minoffset: 104734
maxoffset: 171184 (+66450)
priceChange24H : 14.720716                2023-09-02 17:05:11
nextFundingRate: 0.000037854166666666667   2023-09-02 17:05:41
openInterest   : 607.946                  2023-09-02 17:06:01
trades24H      : 17949                    2023-09-02 17:06:01
volume24H      : 9008342.3964             2023-09-02 17:06:01
effectiveAt    : 2023-09-02T16:30:12.701Z 2023-09-02 16:30:16
effectiveAtHeig: 883105                   2023-09-02 16:30:16
marketId       : 1                        2023-09-02 16:30:16
price          : 1636.875228              2023-09-02 16:30:16
Runtime        : 0:00:00.018604
```

Here you can see the multiple bids (from 1639 to 1645) crossing, and one asks also. But you can also see these bids and asks have lower offsets (below 171k) than the rest. By default, the program logic (with *remove_crossed_prices* to True) assumes the higher offset takes precedence, and removes those prices.