Nicolas11 @ BMT May 17th, 2014 (v1)

Understanding of the structure of a daily .ntd file created by NinjaTrader

Illustration with the example of AAPL in 2014

This is fully based on the work done by **@mrjoe**, **@gomi** and **@dalebru** who decoded .ntf file structure in
https://www.bigmiketrading.com/ninjatrader-programming/7396-ntd-file-specification.html
Thanks to them!

On my computer, NinjaTrader stores the 2014 quotes of AAPL in the following file:
C:\Users\Nicolas\Documents\NinjaTrader 7\db\day**AAPL\2014.Last.ntd**

The beginning of the file is the following:

7b 14 ae 47 e1 7a 84 bf 02 00 00 00 5b 00 00 00 3d 0a d7 a3 70 5d 81 40 0a d7 a3 70 3d 68 81 40 5c 8f c2 f5 28 40 81 40 d7 a3 70 3d 0a 49 81 40 00 80 d3 b6 64 d5 d0 08 03 28 80 00 00 00 00 00 78 91 3e e6 54 04 db 37 00 d6 49 12 79 a2 03 39 fb 03 a7 01 81 04 09 00 e1 4e 19 78 91 42 af a4 02 80 d4 00 ad 26 52 78 62 3d d9 02 a3 0c 01 dd 00 8d 01 7b 78 91 43 1f 06 04 79 75 00 98 61 99 78 91 3d 47 61 03 68 b7 00 a6 5d a8 79 62 03 3c 20 04 eb 03 02 49 00 ce c7 bd 78 62 43 3f 03 53 38 03 69 00 b6 86 c5 78 62 45 fa 02 9c ba 02 3a 00 d6 a7 3d 78 92 40 8a c3 01 42 01 01 00 7d 47 0e 78 91 3e aa 3b 04 86

As further explained below:

- the green part corresponds to the first bar of the year
- the red part corresponds to the second bar
- the violet part corresponds to the third bar
- and so on…

Detailed structure of the daily .ntd file is as follows:

Hexadecimal sequences	Field	Nb of byte(s)	Endian-ness (1)	Type	Value (2)

First record:

(its structure is different from the subsequent ones)

7b 14 ae 47 e1 7a 84 bf	**tick size** (its opposite)	8	LE	IEEE754 double precision 64-bit	0xbf847ae147ae147b = **-0.01**
02 00 00 00	(skipped)	4			
5b 00 00 00	**nb of records**	4	LE	unsigned integer	0x5b = **91**
3d 0a d7 a3 70 5d 81 40	**open**	8	LE	IEEE754 double precision 64-bit	0x40815d70a3d70a3d = **555.68**

Hexadecimal sequences	Field	Nb of byte(s)	Endian-ness (1)	Type	Value (2)
0a d7 a3 70 3d 68 81 40	**high**	8	LE	IEEE754 double precision 64-bit	0x4081683d70a3d70a = **557.03**
5c 8f c2 f5 28 40 81 40	**low**	8	LE	IEEE754 double precision 64-bit	0x40814028f5c28f5c = **552.02**
d7 a3 70 3d 0a 49 81 40	**close**	8	LE	IEEE754 double precision 64-bit	0x4081490a3d70a3d7 = **553.13**
00 80 d3 b6 64 d5 d0 08	**date,** expressed as 10 millionth of a second since Jan. 1st, 0001	8	LE	unsigned integer	0x08d0d564b6d38000 = 635242176000000000 = **Jan. 2nd, 2014**
03 28 80 00 00 00 00 00	**volume**	8	LE	unsigned integer	0x0000000000802803 = **8,398,851**

Second record:

(subsequent records have the same structure)

78 = 0 111 10 00 (3)	**mask** 1	1			

Meaning of mask 1 (after conversion to binary):

Bit 1 (“0”) is ignored.

Bits 2 to 4 (“111”) specify the number of bytes used to store **volume**:

- 001: 1 byte
- 011: 1 byte, but volume to be multiplied by 100
- 011: 1 byte, but volume to be multiplied by 500
- 101: 1 byte, but volume to be multiplied by 1000
- 110: 2 bytes
- 111: 4 bytes
- 010: 8 bytes

Bits 5 and 6 (“10”) specify the number of bytes used to store the **open** price:

- 00: 0 byte (no field)
- 01: 1 bytes
- 10: 2 bytes
- 11: 4 bytes

Bits 7 and 8 (“00”) specify the number of bytes used to store the **delta time**:

- 00 : 0 byte (no field)
- 01: 1 byte
- 10: 2 bytes
- 11: 4 bytes

<table>
<tr><th>Hexadecimal sequences</th><th>Field</th><th>Nb of byte(s)</th><th>Endian-ness (1)</th><th>Type</th><th>Value (2)</th></tr>
<tr><td>91 = 10 01 00 01 (3)</td><td>mask 2</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td colspan="6">Meaning of mask 2 (after conversion to binary):

Bits 1 and 2 (“10”) specify the number of bytes used to store the low:
• 01: 1 byte
• 10: 2 bytes
• 11: 4 bytes

Bits 3 and 4 (“01”) specify the number of bytes used to store the high:
• 01: 1 byte
• 10: 2 bytes
• 11: 4 bytes

Bits 5 and 6 are not used

Bits 7 and 8 (“01”) specify the number of bytes used to store the close:
• 01: 1 byte
• 10: 2 bytes
• 11: 4 bytes</td></tr>
<tr><td>(void)</td><td>delta time
= number of days since previous record (void=1 day)</td><td>(specified by mask 1)</td><td>BE</td><td>unsigned integer</td><td>date = the day after previous record</td></tr>
<tr><td>3e e6</td><td>open delta</td><td>(specified by mask 1)</td><td>BE</td><td>unsigned integer</td><td>0x3ee6 = 16102
from which we can calculate the open</td></tr>
<tr><td colspan="6">The following number shall be subtracted from open delta:
• if stored on 1 byte: 0x80 = 128
• if stored on 2 bytes: 0x4000 = 16384
• if stored on 4 bytes: 0x40000000 = 1073741824
The result is the difference in ticks between the open and the open of the day before.</td></tr>
<tr><td>54</td><td>high delta
= nb of ticks between high and open</td><td>(specified by mask 2)</td><td>BE</td><td>unsigned integer</td><td>0x54 = 84
from which we can calculate the high</td></tr>
<tr><td>04 db</td><td>low delta
= nb of ticks between low and open</td><td>(specified by mask 2)</td><td>BE</td><td>unsigned integer</td><td>0x04db = 1243
from which we can calculate the low</td></tr>
</table>

Hexadecimal sequences	Field	Nb of byte(s)	Endian-ness (1)	Type	Value (2)
37	**close delta** = nb of ticks between low and close	(speci-fied by mask 2)	BE	unsigned integer	0x37 = 55 from which we can calculate the **close**
00 d6 49 12	**volume**	(speci-fied by mask 1)	BE	unsigned integer	0x00d64912 = **14,043,410**

Third record:

(same structure as second record)

79…					

Notes:

(1) Refer to http://fr.wikipedia.org/wiki/Endianness

(2) Conversions can be checked on the following web sites:

- from hexadecimal to unsigned integer: http://www.binaryhexconverter.com/hex-to-decimal-converter
- from hexadecimal to IEEE754 double precision 64-bit: http://www.binaryconvert.com/convert_double.html

(3) Conversion of hexadecimal to binary: http://easycalculation.com/hex-converter.php