QUANTHOUSE

part of S&P CAPITAL IQ

FeedOS API – sample_feed_replay

Release 1.0

FeedOS Team

January 07, 2015

CONTENTS

CHAPTER

ONE

OVERVIEW

`sample_feed_replay` is one of the code sample provided in the C++ FeedOS API.

This sample replays data from a "FeedOS recording/replay server" and prints it in textual form.

This tool outputs the whole feed for a particular feed kind, on all instruments, between two timestamps. It can interact with a server (that stores and streams the feed to be replayed), or directly with files stored on disk.

It is recommended that you write your own *replay and process* program using the C++ API. However you can use `sample_feed_replay` as-is, and parse the text output.

CHAPTER

TWO

USAGE

`sample_feed_replay` is a command line interface (win32 or Linux) tool.

General syntax for invoking the tool is:

```
sample_feed_replay [OPTIONS] {CONNECTION_PARAMETERS} {REPLAY_PARAMETERS}
```

`OPTIONS`

-v	verbose mode
-l	enable log file
-d	debug mode (implies -l)
-nh	don't print header lines
-/	print instrument codes in X/Y format
-instr_code_map FILE	translate numeric instrument codes into custom strings and filter out the output limited to the keys listed in the map
-recv_timestamp	print timestamp of reception, taken by recording device
-epoch_timestamps	print timestamps as 'number of seconds since unix epoch'
-cache_book	don't print delta updates. Reconstruct full book (mutually exclusive with -delta_only)
-delta_only	deprecated, do not use
-trades_only	print only events that are trades
-depth	limit depth of reconstructed book (implies -cache_book)
-L1_extract_daily_values	extract OCHL, volume & asset from L1 stream
-L1_only	for L12 and MUXED: consider only L1 events
-L1_export_histo_daily	export daily values in histo_daily form; requires instr_code_map; mutually exclusive with L1_extract_daily_values

`CONNECTION_PARAMETERS: {-server SERVER PORT LOGIN PASSWORD SRC_ID | -files FEED_DIR SRC_NAME}`

SERVER	IP address of the server
PORT	connection port of the server
LOGIN	user login
PASSWORD	user password
SRC_ID	numeric ID of the source
FEED_DIR	directory where the feed is stored
SRC_NAME	name of the source

`REPLAY_PARAMETERS: FEED_KIND [BEGIN_DATE [END_DATE]]`

FEED_KIND	one of L1, L2, L12, MBO, STATUS, IBAR, MUXED
BEGIN_DATE	"YYYY-MM-DD[HH:MM:SS]" default is '15 minutes ago'
END_DATE	"YYYY-MM-DD[HH:MM:SS]" default is 'now'

Notes:

- CONNECTION_PARAMETERS accepts one of the two ways:
  - -server: to replay feed from a replay storage server
  - -files: to replay feed from the recorded stream saved by a replay storage server
- SRC_ID is an internal code used to identify a source within FeedOS world. They will be provided by Quanthouse along with server IP address and port.

  Special source_id 0 is used to replay an aggregated feed: several sources into a single stream.

  Availability of such feed depends on remote server (type and configuration).
- Ask administrator of the given SERVER about available SRC_IDs / FEED_KINDs
- The instr_map file format is a two-column text file; it materializes a mapping of internal codes to user-defined strings. It can be used to filter some instruments from the replay, or to customize the display.

  To easily generate this file, you can issue the following steps (here raw numerical instrument IDs are mapped to "MIC@LocalCodeStr"):

```
./referential_export -nh -server SERVER PORT LOGIN PASSWORD \
  -f FOSMarketId,LocalCodeStr > ref.txt
awk '{ print $1\"\\t\"$2\"@\"$3 }' ref.txt > numeric_to_localcodestr.txt
```

- Possible values for FEED_KIND are as follows:
  - L1: Trades, best bid/ask and status/values updates
  - L2: Order book updates (aka MBL: Market By Level)
  - L12: Mixed stream of L1 and L2
  - MBO: Market By Order
  - STATUS: Feed status
  - IBAR: N-minutes Intraday Bars (Open, High, Low, Close, Volume)
  - MUXED: Multiplexed (L1, MBL, MBO, STATUS combined)

  Feed kinds available on a server depends on server type and configuration.

CHAPTER

THREE

L1 DATA FORMAT

3.1 Default Mode

Usage:

```
sample_feed_replay -server ip port login pwd 226 L1 "2014-09-12 00:00:00" \
  "2014-09-13 00:00:00"
```

The output consists of text lines showing:

- optional error messages or online help
- a header stating the format used for notifications (option `-nh` prevents it from being output)
- notifications respecting the following format:

```
SI=TradeEvent(SIGNAL) SERVER_TIME MARKET_TIME INSTRUMENT SIGNAL LAST_PRICE
TE=TradeEvent SERVER_TIME MARKET_TIME INSTRUMENT LAST_PRICE TRADE_QTY \
  BID_PRICE BID_QTY ASK_PRICE ASK_QTY [CONTENT_MASK] [FLAGS]
TC=TradeCancelCorrection SERVER_TIME INSTRUMENT CONTENT_MASK Origin: \
  PRICE QTY TRADE_IMPACT_INDICATOR MARKET_TIME TRADE_ID \
  LIST_OF_TRADE_PROPERTIES[ SessionId: SESSION_ID][ Corrected: PRICE QTY \
  TRADE_IMPACT_INDICATOR MARKET_TIME TRADE_ID LIST_OF_TRADE_PROPERTIES] \
  [ CorrectedValues: LIST_OF_VALUES]
VU=ValuesUpdate SERVER_TIME MARKET_TIME INSTRUMENT VALUES...
MS=MarketStatus SERVER_TIME MARKET_TIME MIC STATUS
MN=MarketNews SERVER_TIME MARKET_TIME MIC URGENCY HEADLINE CONTENT
```

N.B. 1 A star ('*') character represents the absence of the specified field in the notification.

N.B. 2 This output, though complete, can make it difficult to separate the different input events, as an event maybe represented with several lines. Further precision on this would require customizing this sample.

We are now going to explain what the output looks like through selected output lines filtered for instrument 692061165.

```
SI 2014-09-12 07:04:59:016 2014-09-12 07:04:59:010 692061165 ChangeBusinessDay *
TE 2014-09-12 07:04:59:016 2014-09-12 07:04:59:010 692061165 * * * * * * y
```

This is the first message of a business day. It consists of a simple signal, `ChangeBusinessDay`, marking the transition between two business days. Each notification containing an event will be output by at least two lines: the SI line expliciting the signals, and the TE line describing the whole content of the notification.

The first timestamp is the server timestamp: the time at which the Quanthouse decoder server received the message from the market.

The second timestamp is the market timestamp: the time at which the event occured as given by the market.

The final star on the SI line indicates that no last price is associated with this signal.

The stars on the TE line indicate that no last price, last trade quantity, bid price, biq quantity, ask price, ask quantity are conveyed in this notification.

The final 'y' on the TE line is the character representation of the content mask of this notification: it describes the content of the notification and also includes signals. Here is the meaning of each possible character:

O	daily open
C	daily close
H	daily high
L	daily low
o	session open
c	session close
h	session high
l	session low
f	off-book trade
y	change business day
e	session event

Note that a daily event implies the equivalent session event if the instrument supports a multisession behavior.

Additionaly a debug version of `sample_feed_replay` have more characters available (that are not necessary to understand the feed but might be useful for investigating issues):

s	trading status presence
b	bid presence
a	ask presence
p	last price presence
q	last trade qty presence
d	daily trait for signal
v	other values presence
n	open next calendar day
x	context tags presence

```
SI 2014-09-12 08:00:00:000 2014-09-12 08:00:00:000 692061165 open *
TE 2014-09-12 08:00:00:000 2014-09-12 08:00:00:000 692061165 * * * * * * oe
VU 2014-09-12 08:00:00:000 2014-09-12 08:00:00:000 692061165 TradingSessionId=-1 \
   TradingStatus=17
```

As above, this notification contains a signal: a session open with no price, bid or ask. But is also contains values update represented by the VU line. Along with the usual line start (type, timestamps, instrument ID), it consists of a list of `TAG_NAME=VALUE`. Here, a session open always contains the ID of the opened session. Moreover the TradingStatus is updated to indicate that trading is active from now on.

```
TE 2014-09-12 08:00:00:052 2014-09-12 07:59:55:757 692061165 * * 85 500 * *
```

This notification contains an update of the best bid price and quantity.

If the price field is '!', the update as to be interpreted as a reset of the best bid (same goes for ask).

If the market provides the information, quantity will consist of both the total quantity and the numbers of orders constituting this quantity. For example: `500@3` means that 3 orders form a total quantity of 500.

```
TE 2014-09-12 08:29:54:526 2014-09-12 08:29:54:526 692061165 101.06 10 * * * * hle \
   TradeID=16288+Q,MARKET_NASDAQ_UTP_SaleCondition=@+To
```

This is a trade: it consists of, at least, a last price and a last trade quantity. Moreover this trade is a session trade, and being the first one of the session, it is the high price and the low price for this session.

It comes with some attached values, displayed at the end of the TE line: a TradeID, and a market-specific tag, SaleCondition.

Values attached to a trade (also called context values) differ from other values (VU lines) to the extent that context values only are meaningful for the trade. They will not "survive" the event they are attached to. Whereas other values persist for the lifetime of the instrument in the snapshot database (even if their values are updated, and can even be reset).

```
SI 2014-09-12 20:00:00:000 2014-09-12 20:00:00:000 692061165 CLOSE  101.67
TE 2014-09-12 20:00:00:000 2014-09-12 20:00:00:000 692061165 101.67 * * * * * C
VU 2014-09-12 20:00:00:000 2014-09-12 20:00:00:000 692061165 InternalDailyClosingPriceType=e
```

This notification conveys the daily close signal and price, along with an update of InternalDailyClosingPriceType.

3.2 L1_extract_daily_values Mode

Usage:

```
sample_feed_replay -L1_extract_daily_values -server ip port login pwd 226 L1 \
  "2014-09-12 00:00:00" "2014-09-16 00:00:00"
```

The output consists of one line per instrument and per trading day. Each line is a quotation summary for the given day and instrument.

Line format:

```
D=L1_DAILY INSTRUMENT BUSINESS_DAY OPEN CLOSE HIGH LOW VOLUME ASSET NB_TRADES OFFBOOK_VOLUME OFFBOOK_
```

Example:

```
D 692061165 2014-09-12  101.22  101.66  102.18  101.08  14210455  1444487508.49 \
  68595 UNQUOTED  UNQUOTED  0
D 692061165 2014-09-15  102.81  101.63  103     101.44  13894101  1419165229.24499 \
  70517 UNQUOTED  UNQUOTED  0
```

Note that 'UNQUOTED' indicates an unset value: here there has been no off-book trade these two days.

3.3 L1_export_histo_daily Mode

This mode requires the use of the `-instr_code_map` option. So, a file has to be created that contains one line for each instrument of interest. Each line reads the instrument ID, a blank separator and any string that can be used for a file name.

Sample instrument code map:

```
$ cat instr_map.txt
692061165 XNAS@AAPL
```

Usage:

```
sample_feed_replay -L1_export_histo_daily -instr_code_map instr_map.txt \
  -server ip port login pwd 226 L1 "2014-09-12 00:00:00" "2014-09-16 00:00:00"
```

The output consists of one file per instrument in instrument code map file. Each file contains a header (except is `-nh` is specified) and one line per trading day. As the previous mode, each line gives a quotation summary for the day and instrument. The format is the same as the one produced by another FeedOS tool: get_histo -daily.

Example:

```
$cat XNAS@AAPL.txt
LOCAL_DATE  DOPEN DCLOSE  DHIGH DLOW  DVOLUME DASSET
2014-09-12  101.22  101.66  102.18  101.08  1.42105e+07 1.44449e+09
2014-09-15  102.81  101.63  103 101.44  1.38941e+07 1.41917e+09
```

'UNQUOTED' string is used wherever one of the value is unset for the day.

CHAPTER

FOUR

MBL DATA FORMAT

4.1 Default Mode

Usage:

```
sample_feed_replay -server ip port login pwd 226 L2 "2014-09-12 00:00:00" \
  "2014-09-13 00:00:00"
```

The output consists of text lines showing:

- optional error messages or online help
- a header stating the format used for notifications (option `-nh` prevents it from being output)
- notifications respecting the following format (MBL kinematic is detailed in an adhoc documentation, for more details please refer to it):

4.1.1 MBLFullRefresh

These events contain a complete snapshot of the multilayer book for an instrument. Each layer will be printed like this:

```
OF SERVER_TIME MARKET_TIME INSTRUMENT LAYER MAX_VISIBLE_DEPTH
_book#  0       BID   53.8500 x    547 @      2 ASK   53.9000 x   1134 @      2
_book#  1       BID   53.8000 x   3776 @      6 ASK   53.9500 x   1393 @      4
_book#  2       BID   53.7500 x  24243 @     12 ASK   54.0000 x  17983 @     15
_book#  3       BID   53.7000 x  26148 @      7 ASK   54.0500 x  23872 @      8
_book#  4       BID   ********************** ASK   54.1000 x   4614 @      7
```

4.1.2 MBLMaxVisibleDepth

```
OV SERVER_TIME MARKET_TIME INSTRUMENT LAYER MAX_VISIBLE_DEPTH

OV null null 621707028 0 10
```

4.1.3 MBLDeltaRefresh

```
OD SERVER_TIME MARKET_TIME INSTRUMENT LAYER ACTION LEVEL PRICE \
   CUMULATED_UNITS NB_ORDERS [CONTINUATION_FLAG]

OD 2014-09-16 06:00:00:063 2014-09-16 06:00:00:063 621712794 0 BidChangeQtyAtLevel \
   0 0 2300 2
```

The existing actions are:

- AllClearFromLevel: delete bid and ask entries from given level
- BidClearFromLevel/AskClearFromLevel: delete bid/ask entries from given level
- BidInsertAtLevel/AskInsertAtLevel: insert given bid/ask entry at given level
- BidRemoveLevel/AskRemoveLevel: delete bid/ask entry at given level
- BidChangeQtyAtLevel/AskChangeQtyAtLevel: change bid/ask quantity at given level
- BidRemoveLevelAndAppend/AskRemoveLevelAndAppend: delete bid/ask entry at given level and append given entry at the end of this side

CONTINUATION_FLAG As long as this flag is set ('C' is displayed), the book cannot be considered consistent. As soon as it is unset (nothing is displayed) the book is consistent.

4.1.4 MBLOverlapRefresh

```
OR SERVER_TIME MARKET_TIME INSTRUMENT LAYER BID ASK

OR 2014-06-19 16:05:34:036 2014-06-19 16:05:34:036 648561381 0 Bid[0,0[(0.0004x115@3) \
   Ask[2,2](0.001x30@1)
```

Update for one side can have 4 different forms:

- absent: no update on this side
-]level]: crop side after level
- [start_level, end_level](limits...): replace limits between start_level and end_level (included) by the given ones and crop after end_level
- [start_level, end_level[(limits...): replace limits between start_level and end_level (included) by the given ones

4.1.5 MBLValuesUpdate

```
VUO SERVER_TIME MARKET_TIME INSTRUMENT LAYER VALUES
```

4.2 Cache-Book Mode

Usage:

```
sample_feed_replay -cache_book -server ip port login pwd 226 L2 \
  "2014-09-12 00:00:00" "2014-09-16 00:00:00"
```

In this mode, events are printed as in the default mode, and additionally each delta event and each overlap event is immediately followed by an output of the reconstructed cache.

Overlap events look a bit different. In default mode:

```
OR 2014-09-16 06:00:00:017 2014-09-16 06:00:00:017 621707077 0 Bid[0,0](45x600@1) Ask]0]
```

In cache_book mode, the side updates are not printed at the end of the OR line, as the output of the book will reflect it anyway. The previous event looks like this:

```
OR 2014-09-16 06:00:00:017 2014-09-16 06:00:00:017 621707077 0 Bid[0,0] Ask]0]
_book#  0       BID   45.0000 x    600 @      1 ASK **************************
```

Delta events look exactly as in default mode:

```
OD 2014-09-16 06:00:00:063 2014-09-16 06:00:00:063 621712794 0 BidChangeQtyAtLevel \
   0 0 2300 2
_book#  0       BID    8.0000 x   2300 @      2 ASK **************************
```

MaxVisibleDepth and FullRefresh events are not followed by the resulting book.

CHAPTER

FIVE

MBO DATA FORMAT

Usage:

```
sample_feed_replay -server ip port login pwd 32 MBO "2013-04-23 00:00:00" \
  "2013-04-24 00:00:00"
```

The output consists of text lines showing:

- optional error messages or online help
- a header stating the format used for notifications (option `-nh` prevents it from being output)
- notifications respecting the following format:

5.1 NewOrder

Insert new order at given level.

```
NO SERVER_TIME MARKET_TIME INSTRUMENT SIDE ORDER_ID PRICE QTY LEVEL

NO 2013-04-23 06:50:00:019.340 2013-04-23 06:50:00:018.977 619410081 B \
   190947528652366063 960 2100 0
```

5.2 ModifyOrder

Modify order at given level (optionnaly move from old_level to new_level).

```
MO SERVER_TIME MARKET_TIME INSTRUMENT SIDE ORDER_ID PRICE QTY OLD_LEVEL LEW_LEVEL

MO 2013-04-23 06:50:00:700.190 2013-04-23 06:50:00:699.950 619432364 A \
   191105351519567894 33.42 25000 0 0
```

5.3 RemoveOneOrder

Remove order at given level.

```
RO SERVER_TIME MARKET_TIME INSTRUMENT SIDE ORDER_ID LEVEL

RO 2013-04-23 06:50:01:300.289 2013-04-23 06:50:01:299.959 619410061 B \
   191105351519567877 0
```

5.4 RemoveAllPreviousOrders

Remove all orders above (and including) given level.

```
RAP SERVER_TIME MARKET_TIME INSTRUMENT SIDE ORDER_ID LEVEL

RAP 2013-04-23 06:50:01:301.000 2013-04-23 06:50:01:300.250 619410061 B \
    1911053515195678778 0
```

5.5 RemoveAllOrders

Remove all orders from the given side(s) ('B', 'A' or '*').

```
RAO SERVER_TIME MARKET_TIME INSTRUMENT SIDE

RAO 2013-04-23 00:30:00:517 2013-04-23 00:30:00:517 619439122 *
```

5.6 Retransmission

Replace complete market sheet by the given one.

```
MBORET SERVER_TIME MARKET_TIME INSTRUMENT BIDsize ASKsize

MBORET 2014-09-16 01:00:00:788 2014-09-16 01:00:00:788 619439122 BIDx0 ASKx0
```

The associated market sheet is not printed.

5.7 ValuesUpdateOneInstrument

MBO can transport other values (like TradingStatus).

```
VUO SERVER_TIME MARKET_TIME INSTRUMENT VALUES...

VUO 2014-09-16 06:30:00:107 2014-09-16 06:30:00:106.557 768318356 TradingStatus=2
```

CHAPTER

SIX

IBAR DATA FORMAT

Usage:

```
sample_feed_replay -files path/to/feed GLOBAL IBAR

IBAR INSTRUMENT SERVER_TIME MARKET_TIME OPEN HIGH LOW CLOSE VOLUME DURATION_SEC \
  NB_POINTS OTHER_VALUES

IBAR 537371695 2014-09-17 13:55:00:000 2014-09-17 13:55:25:088 252.5 252.5 252.5 \
  252.5 27 60 2
IBAR 537371750 2014-09-17 13:55:00:000 null UNQUOTED UNQUOTED UNQUOTED UNQUOTED \
  0 60 0 TradingStatus=5
```

As you can see, bars can transport other values. Thus, they can have no price.

CHAPTER

SEVEN

FEEDSTATUS

Usage:

```
sample_feed_replay -server ip port login pwd 0 STATUS "2014-09-16 00:00:00" \
  "2014-09-17 00:00:00"
```

There are two kinds of notifications:

7.1 FeedStatusUpdate

```
FSU SENDER SEND_TIME FEED_NAME SOURCE_IDS STATE BAD_FLAGS

FSU lse_mit_stream_server/Primary 2014-09-16 00:00:04:872 LSE 33,32 ProbablyNormal
```

7.2 FeedStatusEvent

```
FSE SENDER SEND_TIME FEED_NAME SOURCE_IDS EVENT DETAILS
```