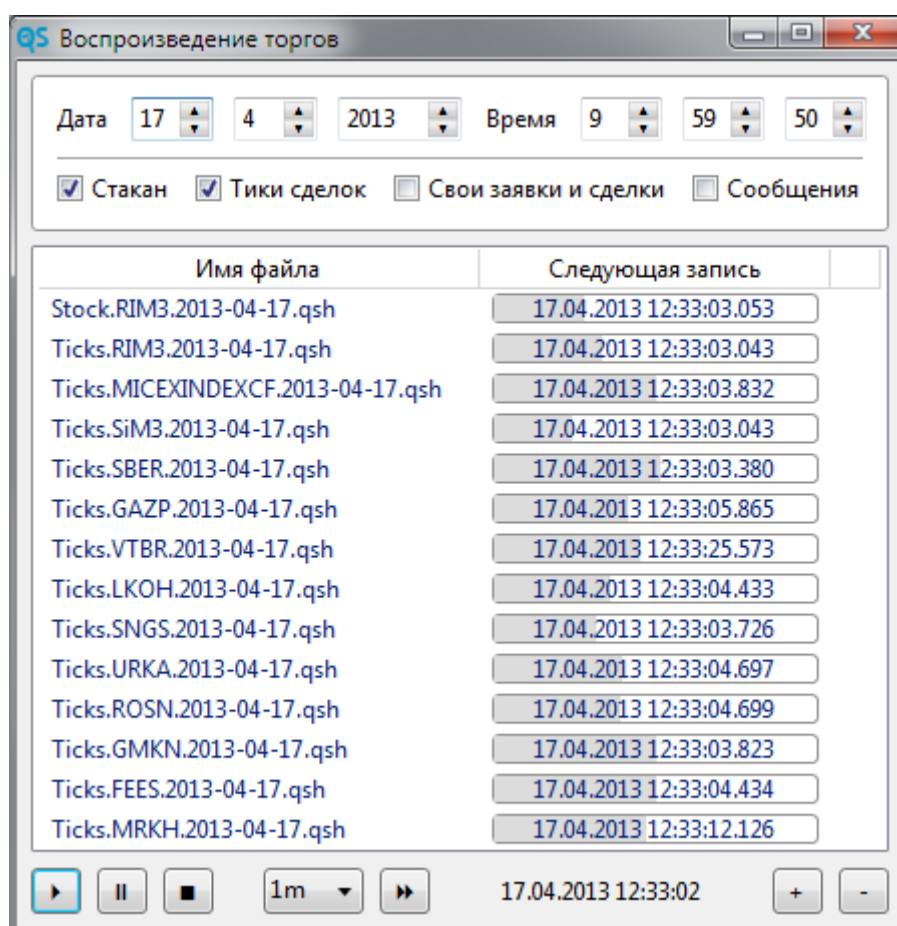


QScalp History Data

СПЕЦИФИКАЦИЯ ФОРМАТА ВЕРСИИ 4



СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ	3
2. ПРЕДСТАВЛЕНИЕ ДАННЫХ	3
2.1. БАЗОВЫЕ ТИПЫ	3
2.2. LEV128 / ULEV128	4
2.3. DATETIME	4
2.4. RELATIVE	4
2.5. STRING	4
2.6. GROWING	4
2.7. GROWDATETIME	5
2.8. ПОЛНЫЙ КОД ИНСТРУМЕНТА	5
3. СТРУКТУРА .QSH ФАЙЛА	5
3.1. ЗАГОЛОВОК ФАЙЛА	5
3.2. ЗАГОЛОВОК ПОТОКА	6
3.3. ЗАГОЛОВОК КАДРА	6
4. СТРУКТУРЫ ДАННЫХ	6
4.1. КОТИРОВКИ БИРЖЕВОГО СТАКАНА	6
4.2. ТИКИ СДЕЛОК	7
4.3. СВОИ ЗАЯВКИ	7
4.4. СВОИ СДЕЛКИ	8
4.5. СООБЩЕНИЯ	8
4.6. ВСПОМОГАТЕЛЬНЫЕ ДАННЫЕ	8
4.7. ПОЛНЫЙ ЖУРНАЛ ЗАЯВОК	9

1. Общие сведения

Формат QScalp History Data (QSH) предназначен для хранения исторических рыночных данных, необходимых для воссоздания хода торгов в реальном масштабе времени. Настоящая спецификация описывает версию 4 данного формата (QScalp 4.6+).

Информация в формате QSH представляется в виде потока бинарных «кадров». В большинстве случаев каждый кадр несет в себе только данные об изменении последнего состояния сущности, которую описывает этот поток.

В одном файле может содержаться до 255-ти различных информационных потоков. Если количество потоков более одного, каждый кадр снабжается информацией о том, к какому потоку он принадлежит. При этом размер кадра увеличивается на 1 байт. Таким образом, для достижения максимальной компактности рекомендуется использовать один поток на файл.

Информационные потоки могут быть следующих типов:

- котировки биржевого стакана (Quotes);
- тики сделок (Deals);
- собственные заявки (OwnOrders);
- собственные сделки (OwnTrades);
- сообщения (Messages);
- вспомогательные данные (AuxInfo);
- полный журнал заявок (OrdLog).

Независимо от количества потоков в файле, каждый кадр снабжается сквозным штампом времени, который может соответствовать моменту получения или записи данной порции информации. Ситуация, когда последующий штамп времени имеет меньшее значение, чем предыдущий, является допустимой, но некорректной с точки зрения логики потоков.

Файлы в формате QSH после окончания записи могут быть дополнительно сжаты с помощью gzip-компрессии. При этом расширение файла может быть оставлено первоначальным (.qsh) или же изменено путем добавления к нему соответствующего окончания (.qsh.gz). Сжатые файлы воспроизводятся наравне с несжатыми, а наличие компрессии определяется путем анализа заголовка файла.

2. Представление данных

2.1. Базовые типы

Тип	Описание
byte	беззнаковое целое число длиной 8 бит
uint16	беззнаковое целое число длиной 16 бит
uint32	беззнаковое целое число длиной 32 бита
int64	знаковое целое число длиной 64 бита
double	число с плавающей запятой двойной точности длиной 64 бита

2.2. Leb128 / ULeb128

Тип **Leb128** (Little Endian Base 128) / **ULeb128** (Unsigned Little Endian Base 128) является общепринятым форматом кодирования чисел переменной длины. Описание данного формата доступно по ссылке: <https://en.wikipedia.org/wiki/LEB128>.

Исходное кодируемое число имеет следующий базовый тип:

- **int64** для Leb128;
- **uint32** для ULeb128.

2.3. DateTime

Тип **DateTime** представляет собой число, показывающее количество 100-наносекундных интервалов, которые прошли с полночи 00:00:00, 1 января 0001 года. Соответствует свойству Ticks структуры DateTime из .NET версии 4. Сохраняется в поле типа **int64**.

2.4. Relative

Тип **Relative** представляет собой число, закодированное в формате **Leb128**, показывающее разность между текущим значением и предыдущим. Первая разность берется относительно нуля.

2.5. String

Тип **String** является комплексным и состоит из следующих компонентов:

Тип	Описание
ULeb128	длина последующего массива байт
byte[]	массив байт, содержащий строку в кодировке UTF8

2.6. Growing

Тип **Growing** является комплексным и состоит из одного или двух компонентов:

Тип	Описание
ULeb128	разность между текущим и предыдущим значением; если данная разность меньше нуля или больше 268435454, в этом поле указывается число 268435455, а значение разности указывается в следующем поле
Leb128	разность между текущим и предыдущим значением, если предыдущее поле содержит число 268435455; в ином случае данное поле отсутствует

Для самого первого значения разность вычисляется относительно нуля.

2.7. GrowDateTime

Тип **GrowDateTime** представляет собой количество миллисекунд, которые прошли с полночи 00:00:00, 1 января 0001 года. Соответствует свойству `Ticks` структуры `DateTime` из .NET версии 4, деленному на константу `TimeSpan.TicksPerMillisecond`. Сохраняется в поле типа **Growing**.

2.8. Полный код инструмента

Полный код инструмента представляет собой строку следующего формата:

```
CONNECTOR:TICKER:AUXCODE:NUMCODE:STEP
```

- **CONNECTOR** – идентификатор коннектора/подключения, через который выполняется получение данных по инструменту;
- **TICKER** – основной код инструмента (например, RIU3);
- **AUXCODE** – вспомогательный код инструмента (например, SPBFUT, CME), в некоторых случаях отсутствует;
- **NUMCODE** – цифровой код инструмента, в некоторых случаях отсутствует;
- **STEP** – минимальный шаг цены инструмента.

3. Структура .qsh файла

Структура файла является следующей:

- заголовок файла
- заголовок потока 1
- заголовок потока 2 (при наличии)
- ...
- заголовок потока N (при наличии)
- заголовок кадра 1
- данные кадра 1
- заголовок кадра 2
- данные кадра 2
- ...
- заголовок кадра N
- данные кадра N

3.1. Заголовок файла

Тип	Описание
byte[]	сигнатура файла = «QScalp History Data» (только символы UTF8, без нуля в конце)
byte	мажорная версия формата файла = 4
String	имя приложения, с помощью которого записан данный файл
String	произвольный пользовательский комментарий
DateTime	дата и время начала записи файла (UTC)

Тип	Описание
byte	количество информационных потоков в файле

3.2. Заголовок потока

Тип	Описание
byte	идентификатор потока: <ul style="list-style-type: none"> • Quotes = 0x10 • Deals = 0x20 • OwnOrders = 0x30 • OwnTrades = 0x40 • Messages = 0x50 • AuxInfo = 0x60 • OrdLog = 0x70
String	полный код инструмента, которому соответствует поток; для потока «Messages» отсутствует

3.3. Заголовок кадра

Тип	Описание
GrowDateTime	штамп времени (UTC)
byte	номер потока (начинающийся с нуля) в соответствии со списком заголовков потоков, которому принадлежит кадр; указывается, только если файл содержит более одного потока

4. Структуры данных

4.1. Котировки биржевого стакана

Тип	Описание
Leb128	количество котировок в данном кадре
Массив котировок:	
Relative	цена в тиках (шагах цены)
Leb128	объем: положительный для котировок «ask», отрицательный для котировок «bid», нулевой, если по данной цене нет заявок

Первый кадр в потоке содержит полный стакан котировок. Все последующие только изменения. Цена в котировке сохраняется относительно цены предыдущей котировки.

4.2. Тики сделок

Тип	Описание
byte	<p>набор битовых флагов:</p> <ul style="list-style-type: none"> • биты 0..1 – указывают направление сделки: 0 – неизвестное направление, 1 – покупка, 2 – продажа, 3 – зарезервировано; <p>последующие биты устанавливаются в 1 в случае присутствия в кадре соответствующего поля:</p> <ul style="list-style-type: none"> • бит 2 – биржевые дата и время сделки; • бит 3 – номер сделки в торговой системе; • бит 4 – номер заявки, по которой была совершена данная сделка; • бит 5 – цена сделки; • бит 6 – объем сделки; • бит 7 – открытый интерес по инструменту после совершения сделки.
GrowDateTime	биржевые дата и время сделки
Growing	номер сделки в торговой системе
Relative	номер заявки, по которой совершена данная сделка
Relative	цена сделки в тиках (шагах цены)
Leb128	объем сделки
Relative	открытый интерес по инструменту после совершения сделки

Наличие тех или иных данных в кадре определяется его первым байтом, который представляет собой набор битовых флагов. Каждый последующий кадр содержит эту информацию, только если она изменилась. Соответственно, если каких-либо данных в кадре нет, следует использовать информацию от предыдущей сделки/кадра. Первоначально все поля инициализируются нулями.

4.3. Свои заявки

Тип	Описание
byte	<p>набор битовых флагов:</p> <ul style="list-style-type: none"> • бит 0 – все заявки были сняты (в этом случае остальные поля в кадре отсутствуют); • бит 1 – заявка активна, иначе – снята/исполнена; • бит 2 – заявка является внешней (биржевой/брокерской); • бит 3 – признак стоп-заявки.
Leb128	номер заявки
Leb128	цена в заявке в тиках (шагах цены)
Leb128	остаток в заявке (положительный для покупки, отрицательный для продажи)

4.4. Свои сделки

Тип	Описание
GrowDateTime	биржевое время сделки
Relative	номер сделки в торговой системе
Relative	номер заявки, по которой совершена данная сделка
Relative	цена сделки в тиках (шагах цены)
Leb128	объем сделки (положительный для покупки, отрицательный для продажи)

4.5. Сообщения

Тип	Описание
DateTime	локальное время сообщения
byte	тип сообщения: <ul style="list-style-type: none"> • 0x01 – информационное; • 0x02 – предупреждение; • 0x03 – ошибка.
String	текст сообщения

4.6. Вспомогательные данные

Тип	Описание
byte	набор битовых флагов; установленный бит означает присутствие в кадре указанной информации: <ul style="list-style-type: none"> • бит 0 – биржевое время обновления данных; • бит 1 – суммарный объем котировок «ask»; • бит 2 – суммарный объем котировок «bid»; • бит 3 – количество открытых позиций; • бит 4 – цена последней сделки инструменту; • бит 5 – информация о сессии: верхний лимит цены, нижний лимит цены, гарантийное обеспечение; • бит 6 – курс пересчета пунктов инструмента в денежные единицы; • бит 7 – сообщение торговой системы.
GrowDateTime	биржевое время обновления данных
Relative	суммарный объем котировок «ask»
Relative	суммарный объем котировок «bid»
Relative	количество открытых позиций (открытый интерес)
Relative	цена последней сделки по инструменту в тиках

Тип	Описание
Leb128	верхний лимит цены в тиках
Leb128	нижний лимит цены в тиках
double	гарантийное обеспечение в денежных единицах
double	курс пересчета пунктов инструмента в денежные единицы
String	сообщение торговой системы

Наличие тех или иных данных в кадре определяется его первым байтом, который представляет собой набор битовых флагов. Каждый последующий кадр содержит эту информацию, только если она изменилась. Соответственно, если каких-либо данных в кадре нет, следует использовать информацию от предыдущего кадра. Первоначально все поля инициализируются нулями.

Бит 5 – информация о сессии – указывает на наличие в кадре сразу трех полей: верхний лимит цены, нижний лимит цены и гарантийное обеспечение. Если он не установлен, все три поля в кадре отсутствуют.

4.7. Полный журнал заявок

Тип	Описание
byte	<p>набор битовых флагов; установленный бит означает присутствие в кадре указанной информации:</p> <ul style="list-style-type: none"> • бит 0 – биржевое время обновления данных; • бит 1 – номер заявки в торговой системе; • бит 2 – цена в заявке; • бит 3 – количество инструмента в данной операции; • бит 4 – остаток в заявке; • бит 5 – номер сделки, в которую сведена заявка; • бит 6 – цена сделки, в которую была сведена заявка; • бит 7 – открытый интерес после заключения сделки.
uint16	<p>набор битовых флагов, характеризующих данную операцию в соответствии со спецификацией Plaza II:</p> <ul style="list-style-type: none"> • бит 0 – при получении данной записи поле ReplAct не было равно нулю (NonZeroReplAct); • бит 1 – данная запись получена с новым идентификатором сессии или после сообщения смены номера жизни потока (FlowStart); • бит 2 – новая заявка (Add); • бит 3 – заявка сведена в сделку (Fill); • бит 4 – покупка (Buy); • бит 5 – продажа (Sell); • бит 6 – запись получена из архива торговой системы (Snapshot); • бит 7 – котировочная (Quote); • бит 8 – встречная (Counter); • бит 9 – внесистемная (NonSystem);

Тип	Описание
	<ul style="list-style-type: none"> • бит 10 – запись является последней в транзакции (EndOfTransaction); • бит 11 – заявка fill-or-kill (FillOrKill); • бит 12 – запись является результатом операции перемещения заявки (Moved); • бит 13 – запись является результатом операции удаления заявки (Canceled); • бит 14 – запись является результатом группового удаления (CanceledGroup); • бит 15 – признак удаления остатка заявки по причине кросс-сделки (CrossTrade).
GrowDateTime	биржевое время
Growing / Relative	биржевой номер заявки; если установлен флаг «Add» поле имеет тип Growing, иначе поле имеет тип Relative и сохраняется относительно последнего значения данного поля для записи с флагом «Add»
Relative	цена в заявке в тиках (шагах цены)
Leb128	объем операции
Leb128	остаток в заявке после операции; указывается только для записей с флагом «Fill»
Growing	номер сделки, в которую была сведена заявка; указывается только для записей с флагом «Fill»
Relative	цена сделки в тиках; указывается только для записей с флагом «Fill»
Relative	значение открытого интереса после сделки; указывается только для записей с флагом «Fill»

Наличие тех или иных данных в кадре определяется его первым байтом, который представляет собой набор битовых флагов. Каждый последующий кадр содержит эту информацию, только если она изменилась. Соответственно, если каких-либо данных в кадре нет, следует использовать информацию от предыдущего кадра, если она применима к данной записи. Первоначально все поля инициализируются нулями.

Остаток в заявке, номер сделки, цена сделки и открытый интерес указываются только для записей с флагом «Fill». При чтении данных для записей без данного флага следует в этих полях задать нулевые значения. Для записей с флагом «Add» при этом остаток в заявке следует задать равным объему операции.

Данный поток является представлением таблицы orders_log потока FORTS_ORDLOG_REPL на основе спецификации шлюза Plaza II (ftp://ftp.moex.com/pub/ClientsAPI/Spectra/Docs/p2gate_ru.pdf).